MICROPROCESSORS and PERIPHERALS

DATA BOOK

THOMSON COMPONENTS
MOSTEK

MICROPROCESSORS and PERIPHERALS
DATA BOOK

THOMSON
MOSTEK
1987

CONTENTS

CONTENTS

THOMSON - MOSTEK
TECHNICAL SUPPORT, PRODUCT MARKETING

FOR INTEGRATED CIRCUITS:
(Corporate Headquarters)
1310 Electronics
Carrollton, TX 75006
214/466-6000
TWX 910-860-5437

FOR DISCRETE DEVICES AND RF & MICROWAVE TRANSISTORS:
16 Commerce Drive
Montgomeryville, PA 18936-1002
215/362-8500
FAX 215/362-1293

FOR PASSIVE COMPONENTS:
6203 Variel Avenue, Unit A
P.O. Box 4051
Woodland Hills, CA 91367
818/887-1010
FAX 818/702-0725

FOR SPECIAL PRODUCTS:
301 Route 17 North
Rutherford, NJ 07070
201/438-2300
FAX 201/438-1774

U.S. AND CANADIAN SALES OFFICES:

WESTERN AREA:

Thomson Components - Mostek Corporation
2540 Mission College Blvd.
Suite 104
Santa Clara, CA 95054
408/970-8585
FAX 408-970-8737

Thomson Components - Mostek Corporation
18004 Skypark Circle
Suite 140
Irvine, CA 92714
714/250-0455
FAX 714/261-1505

Thomson Components - Mostek Corporation
6203 Variel Ave.
Unit A, P.O. Box 4051
Woodland Hills, CA 91367
818/887-1010
FAX 818/702-0725

Thomson Components - Mostek Corporation
1107 North East 45th St.
Suite 411
Seattle, WA 98105
206/632-0245
FAX 206/633-5413

Thomson Components - Mostek Corporation
601 South Bowen St.
Longmont, CO 80501
303/449-9000
FAX 303/651-7976

Thomson Components - Mostek Corporation
7950 East Redfield Rd.
Suite 160
Scottsdale, AZ 85260
602/998-1580
FAX 602/483-2303

Thomson Components - Mostek Corporation
7155 SW Varns St.
Tigard, OR 97223-8057
503/620-5517
FAX 503/639-3591

FOR ALL OTHER COUNTRIES
Thomson Semiconducteurs
43, Avenue de L'Europe
78140 Vélizy-Villacoublay/France
Tél: (1) 39 46 97 19/Télex: 204 780 F
or contact Corporate Headquarters

CENTRAL AREA:

Thomson Components - Mostek Corporation
1310 Electronics
MS1137
Carrollton, TX 75006
214/466-8844
TWX 910-860-5437

Thomson Components - Mostek Corporation
6100 Green Valley Drive
Suite 130
Bloomington, MN 55438
612/831-2322
FAX 612/831-8195

Thomson Components - Mostek Corporation
1827 Walden Office Square
#430
Schaumburg, IL 60173
312/397-6550
FAX 312/397-4066

Thomson Components - Mostek Corporation
3215 Steck Ave.
Suite 202
Austin, TX 78758
512/451-4061
TWX 910-874-2007

Thomson Components - Mostek Corporation
26677 W. 12 Mile Rd.
Suite #141
Southfield, MI 48034
313/354-5840
FAX 313/354-3370

CANADA

Thomson CSF Canada
Semiconductor Division
1000 Sherbrooke West
Suite 2340
Montreal, Quebec H3A 3G4
514/288-4148
FAX 514/288-8987

Thomson Components - Mostek Corporation
44 Rosebud Ave.
Brampton, Ontario L6X 2W5
416/454-5252
FAX 416/454-4328

EASTERN AREA:

Thomson Components - Mostek Corporation
83 Cambridge Street
Suite 2A
Burlington, MA 01803
617/273-3310
FAX 617/272-2467

Thomson Components - Mostek Corporation
The Pavilions at Greentree
Route #73, Suite 101
Marlton, NJ 08053
609/596-9200
FAX 609/424-7437

Thomson Components - Mostek Corporation
4414 Evangel Cr. #C
Huntsville, AL 35816
205/830-9036
FAX 205/830-9038

Thomson Components - Mostek Corporation
387 Hooker Avenue
Office No. 2
Poughkeepsie, NY 12603
914/454-8813
FAX 914/454-1320

Thomson Components - Mostek Corporation
5890 Sawmill Rd.
Suite 204
Dublin, Ohio 43017
614/761-0676
FAX 614/761-2305

Thomson Components - Mostek Corporation
6045 Atlantic Blvd.
Suite 104
Norcross, GA 30071
404/662-1588
FAX 404/662-1561

U.S. AND CANADIAN REPRESENTATIVES

CONNECTICUT
Scientific Components*
1185 S. Main St.
Cheshire, CT 06410
(203) 272-2963
FAX (203) 271-3048

FLORIDA
Lawrence Associates*
5021 N. Dixie Hwy.
Boca Raton, FL 33431
(305) 368-7373

Sales Engineering Concepts, Inc.
926 Great Pond Dr.
Suite 2002
Altamonte Springs, FL 32714
(305) 682-4800
FAX (305) 682-6491

Sales Engineering Concepts, Inc.*
1000 S. Federal Hwy.
Suite 204
Deerfield Beach, FL 33441
(305) 426-4601
TWX 510-600-7740

ILLINOIS
Eagle Technical Sales, Inc.
1805 B. Hicks Rd.
Rolling Meadows, IL 60008
(312) 991-0700

INDIANA
M/S Sales Associates, Inc.*
7319 W. Jefferson Blvd.
Ft. Wayne, IN 46804
(219) 436-3023
FAX (219) 436-3026

M/S Sales Associates, Inc.
1425 E. 86th St.
Indianapolis, IN 46240
(317) 257-8916

IOWA
Rep Associates*
980 Arica Ave.
Marion, IA 52302
(319) 373-0152

KANSAS
Rush & West Associates
107 N. Chester Street
Olathe, KS 66061
(913) 764-2700
TWX 910-380-8110

MARYLAND
Tri-Mark Inc.*
1410 Crain Hwy. NW
Glen Burnie, MD 21061
(301) 761-6000
FAX (301) 761-6006

MASSACHUSETTS
A/D Nova Sales, Inc.*
83 Cambridge St.
Suite 2D
Burlington, MA 01803
(617) 270-9600
FAX (617) 272-2467

Conti-Younger Assoc.*
12 Blanchard Rd.
Burlington, MA 01803
(617) 273-1583
FAX (617) 270-0301

MICHIGAN
Electronic Sources, Inc.*
8014 W. Grand River
Suite 6
Brighton, MI 48116
(313) 227-3598
FAX (313) 227-5655

Centech, Inc.*
10312 E. 63rd Terrace
Raytown, MI 64133
(816) 358-8100

MINNESOTA
Horizon*
8053 East Bloomington Freeway
Bloomington, MN 55420
(612) 884-6515
FAX (612) 888-3073

MISSOURI
Rush & West Associates*
2170 Mason Rd.
St. Louis, MO 63131
(314) 965-3322
TWX 910-752-653
TELEX 752 653

NEW JERSEY
Tritek Sales, Inc.*
21 E. Euclid Ave.
Haddonfield, NJ 08033
(609) 429-1551
TWX 215-627-0149 (Philadelphia Line)
TWX 710-896-0881

NE Components*
189-191 Godwin Ave.
Wyckoff, NJ 07481
(201) 848-1100

NEW YORK
Empire Technical Assoc.*
33 West State St.
Suite 211B
Binghamton, NY 13901
(607) 772-0651

Empire Technical Assoc.
1551 E. Genesse St.
Skaneateles, NY 13152
(315) 685-5703

GT Sales*
34 Grand Blvd.
Brentwood, NY 11717
(516) 231-0270
FAX (516) 273-1247

OHIO
Five Star Electronics*
6200 S.O.M. Center Road
Suite B 21
Solon, OH 44139
(216) 349-1611

Tom Mulligan Co.*
166 N.W. Professional Plaza
Columbus, OH 43220
(614) 457-2242

PENNSYLVANIA
M. Lader Company*
456 Germantown Pike
Lafayette Hill, PA 19444
(215) 825-3177

WISCONSIN
Heartland Technical Marketing*
3846 Wisconsin Ave.
Milwaukee, WI 53208
(414) 931-0606

CANADA
Solution Electronic Sales*
100A 3380 Maquinna Dr.
Vancouver, B.C. V5S 4C9
(604) 438-0679

* Home Office

U.S. AND CANADIAN DISTRIBUTORS

ALABAMA
Marshall Industries
3313 S. Memorial Pkwy.
Huntsville, AL 35801
(205) 881-9235

Quality Components, S.E.
4900 University Square #207
Huntsville, AL 35817
(205) 830-1881

Pioneer Technologies Group
4825 University Square
Huntsville, AL 35805
(205) 837-9300

Schweber Electronics
4930-A Corporate Drive
Huntsville, AL 35805
(205) 895-0480

ARIZONA
Kierulff Electronics
4134 E. Wood Street
Phoenix, AZ 85040
(602) 437-0750

Marshall Industries
835 West 22nd St.
Tempe, AZ 85282
(602) 968-6181

Schweber Electronics
11049 North 23rd Drive
Suite 100
Phoenix, AZ 85029
(602) 997-4874

ARKANSAS
See Oklahoma

CALIFORNIA
Integrated Electronics Corp.
7000 Franklin Blvd., Suite 625
Sacramento, CA 95823
(916) 424-5297

Integrated Electronics Corp.
2170 Paragon Drive
San Jose, CA 95131
(408) 435-1000

ITAL Sales
15405 Proctor Avenue
City of Industry, CA 91745
(818) 968-8515

Kierulff Electronics
9800 Variel Avenue
Chatsworth, CA 91311
(818) 407-2500

Kierulff Electronics
5650 Jillson St.
Los Angeles, CA 90040
(213) 725-0325

Kierulff Electronics
8797 Balboa Avenue
San Diego, CA 92123
(619) 278-2112

Kierulff Electronics
1180 Murphy Avenue
San Jose, CA 95131
(408) 971-2600

Kierulff Electronics
14101 Franklin Avenue
Tustin, CA 92680
(714) 731-5711

Marshall Industries
One Morgan
Irvine, CA 92715
(714) 458-5395

Marshall Industries
9710 DeSoto Avenue
Chatsworth, CA 91311
(818) 407-0101

Marshall Industries
3039 Kilgore Ave., #140
Rancho Cordova, CA 95670
(916) 635-9700

Marshall Industries
10105 Carroll Canyon Rd.
San Diego, CA 92131
(619) 578-9600

Marshall Industries
336 Los Coches St.
Milpitas, CA 95035
(408) 943-4600

Schweber Electronics
21139 Victory Blvd.
Conoga Park, CA 91303
(818) 999-4702

Schweber Electronics
1225 West 190th Street
Suite 360
Gardena, CA 90248
(213) 327-8409

Schweber Electronics
17822 Gillette Avenue
Irvine, CA 92714
(714) 863-0264
FAX (714) 863-0200 (X500)

Schweber Electronics
1771 Tribute Rd. Suite B
Sacramento, CA 95815
(916) 929-9732
FAX (916) 929-5608

Schweber Electronics
6750 Nancy Ridge Drive
San Diego, CA 92121
(619) 450-0454
TWX 910-335-1155

Schweber Electronics
90 E. Tasman Drive
San Jose, CA 95131
(408) 946-7171

Zeus Components
1130 Hawk Circle
Anaheim, CA 92807
(714) 632-6880
TWX 910-591-1696
FAX (714) 630-8770

Zeus Components
1580 Old Oakland Road
Suite C205/C206
San Jose, CA 95131
(408) 998-5121
TWX 408-628-96083
FAX (408) 998-0285

COLORADO
Integrated Electronics Corp.
5750 N. Logan Street
Denver, CO 80216
(303) 292-6121

Kierulff Electronics
7060 South Tuscon Way
Englewood, CO 80112
(303) 790-4444

Marshall Industries
7000 North Broadway
Denver, CO 80221
(303) 427-1818

Schweber Electronics
8955 E. Nicholas, Bldg. 2
Englewood, CO 80221
(303) 799-0258

CONNECTICUT
Greene-Shaw
1475 Whalley Avenue
New Haven, CT 06525
(203) 397-0710
TWX 92-2498

Marshall Industries
20 Sterling Drive
Barnes Industrial Park, N.
P.O. Box 200
Wallingford, CT 06492-0200
(203) 265-3822

Pioneer-Standard
112 Main Street
Norwalk, CT 06851
(203) 853-1515
TWX 710-468-3373
FAX (203) 838-9901

Schweber Electronics
Commercial Industrial Park
Finance Drive
Danbury, CT 06810
(203) 748-7080
TWX 710-456-9405

DELAWARE
See New Jersey
Pennsylvania

FLORIDA
All American Semiconductor
16251 N.W. 54th Avenue
Miami, FL 33014
(305) 621-8282
800-327-6237

Marshall Industries
4205 34th St., S.W.
Orlando, FL 32811
(305) 841-1878
(305) 841-1878

Pioneer Technologies Group
337 S. North Lake, #1000
Altamonte Springs, FL 32701
(305) 834-9090
TWX 810-853-0284

Pioneer Technologies Group
674 S. Military Trail
Deerfield Beach, FL 33441
(305) 428-8877
TWX 510-955-9653

Schweber Electronics
215 North Lake Blvd.
Altamonte Springs, FL 32701
(305) 331-7555
TWX 510-954-0304

Schweber Electronics
3665 Park Central Blvd. N.
Pompano Beach, FL 33064
(305) 977-7511
TWX 510-954-0304

Zeus Components
1750 West Broadway
Suite 114
Oviedo, FL 32765
(305) 365-3000
TWX 910-380-7430
FAX (305) 365-2356

GEORGIA
Dixie Electronics
1234 Gordon Park Road
Augusta, GA 30901
(404) 722-2055

Pioneer Technologies Group
3100 F. Northwoods Place
Norcross, GA 30071
(404) 448-1711
TWX 810-766-4515

Quality Components
6145 N. Belt Parkway #B
Norcross, GA 30071
(404) 449-9508
TWX 510-601-5297
629-32421

Schweber Electronics
303 Research Drive
Suite 210
Norcross, GA 30092
(404) 449-9170
TWX 810-766-1592

ILLINOIS
Advent Electronics
7110-16 N. Lyndon St.
Rosemont, IL 60018
(312) 297-6200

Kierulff Electronics
1140 W. Thorndale Ave.
Itasca, IL 60143
(312) 250-0500

Marshall Industries
1261 Wiley Road
#F
Schaumburg, IL 60195
(312) 490-0155

Pioneer-Standard
1551 Carmen Drive
Elk Grove Village, IL 60007
(312) 437-9680

Schweber Electronics
904 Cambridge Drive
Elk Grove Village, IL 60007
(312) 364-3750
TWX 910-222-3453

U.S. AND CANADIAN DISTRIBUTORS

INDIANA
Advent Electronics
8446 Moller Road
Indianapolis, IN 46268
(317) 872-4910
TWX 810-341-3228

Marshall Industries
6990 Corporate Dr.
Indianapolis, IN 46278
(317) 297-0483

Pioneer-Standard
6408 Castleplace Drive
Indianapolis, IN 46250
(317) 849-7300
TWX 810-260-1794

IOWA
Advent Electronics
682 58th Avenue, Ct. SW
Cedar Rapids, IA 52404
(319) 363-0221
TWX 910-525-1337

Schweber Electronics
5270 North Park Place, NE
Cedar Rapids, IA 52402
(319) 373-1417

KANSAS
Marshall Industries
8321 Melrose Dr.
Lenexa, KS 66214
(913) 492-3121

Schweber Electronics
10300 West 103rd Street
Suite 200
Overland Park, KS 66214
(913) 492-2922

KENTUCKY
See Indiana

LOUISIANA
See Texas

MAINE
See Massachusetts

MARYLAND
Marshall Industries
8445 Helgerman Court
Gaithersburg, MD 20877
(301) 840-9450

Pioneer Technologies Group
9100 Gaither Road
Gaithersburg, MD 20877
(301) 921-0660
TWX 710-828-0545

Schweber Electronics
9330 Gaither Road
Gaithersburg, MD 20877
(301) 840-5900
TWX 710-828-9749

Zeus Components
8930-A Route 108
Columbia, MD 21045
(301) 997-1118
TWX 910-380-3554
FAX (301) 964-9784

MASSACHUSETTS
Greene-Shaw
70 Bridge Street
Newton, MA 02195
(617) 969-8900
TWX 92 2498

Kierulff Electronics
13 Fortune Drive
Billerica, MA 01821
(617) 667-8331

Lionex Corporation
36 Jonspin Road
Wilmington, MA 01887
(617) 657-5170
FAX (617) 657-6008

Marshall Industries
One Wilshire Road
Burlington, MA 01803
(617) 272-8200

Pioneer-Standard
44 Hartwell Ave.
Lexington, MA 02173

Schweber Electronics
25 Wiggins Avenue
Bedford, MA 01730
(617) 275-5100
TWX 710-326-0268

Zeus Components
429 Marrett Road
Lexington, MA 02173
(617) 863-8800
TWX 710-326-7604
FAX (617) 863-8807

MICHIGAN
Advent Electronics
24713 Crestview Ct.
Farmington Hills, MI 48018
(313) 477-1650

Pioneer-Standard
4505 Broadmoor Avenue SE
Grand Rapids, MI 49508
(616) 698-1800
TWX 510-600-8456

Pioneer-Standard
13485 Stamford
Livonia, MI 48150
(313) 525-1800
TWX 810-242-3271

Schweber Electronics
12060 Hubbard Ave. CN3306
Livonia, MI 48150
(313) 525-8100
TWX 810-242-2983

MINNESOTA
Kierulff Electronics
7667 Cahill Road
Edina, MN 55435
(612) 941-7500

Marshall Industries
3800 Annapolis Lane
Plymouth, MN 55441
(612) 559-2211

Pioneer Standard
10203 Bren Road East
Minnetonka, MN 55343
(612) 935-5444
TWX 910-576-2738

Schweber Electronics
7424 W. 78th Street
Edina, MN 55435
(612) 941-5280
TWX 910-576-3167

MISSISSIPPI
See Texas

MISSOURI
Kierulff Electronics
11804 Borman Drive
St. Louis, MO 63146
(314) 997-4956
TWX 910-762-0721

Schweber Electronics
502 Earth City Expressway
Suite 203
Earth City, MO 63045
(314) 739-0526
TWX 43-4065

MONTANA
See California

NEBRASKA
See Iowa

NEW HAMPSHIRE
Schweber Electronics
Bedford Farms Bldg. #2
Manchester, NH 03102
(603) 625-2250
TWX 710-220-7572
FAX (603) 625-5710

NEW JERSEY
Kierulff Electronics
37 Kulick Road
Fairfield, NJ 07006
(201) 575-6750

Marshall Industries
101 Fairfield Rd.
Fairfield, NJ 07006
(201) 882-0320

Pioneer-Standard
45 Route 46
Pine Brook, NJ 07058
(201) 575-3510
TWX 710-734-4382

Schweber Electronics
18 Madison Road
Fairfield, NJ 07006
(201) 227-7880
TWX 710-734-4305

Solid State
46 Farrand Street
Bloomfield, NJ 07003
(201) 429-8700
TWX 710-994-4780
FAX (201) 429-8683

NEW YORK
Add Electronics
7 Adler Drive
E. Syracuse, NY 13057
(315) 437-0300

Add Electronics
7375 Pittsford-Victor Rd.
Victor, NY 14564

Marshall Industries
129 Brown St.
Johnson City, NY 13790

Marshall Industries
1280 Scottsville Rd.
Rochester, NY 14624

Nu-Horizons Electronics
6000 New Horizons Blvd.
N. Amityville, NY 11701
(516) 226-6000

Pioneer-Standard
840 Fairport Park
Fairport, NY 14450
(716) 381-7070
TWX 510-253-7001
FAX (716) 381-5955

Pioneer-Standard
1806 Vestal Pkwy. East
Vestal, NY 13850
(607) 748-8211
TWX 510-252-0893

Pioneer-Standard
Crossways Park West
Woodbury, NY 11797
(516) 921-8700
TWX 510-221-2184
FAX (516) 921-2143

Schweber Electronics
3 Townline Circle
Rochester, NY 14623
(716) 424-2222
TWX 710-541-0601

Schweber Electronics
Jericho Turnpike
Westbury, NY 11590
(516) 334-7474
TWX 510-220-1365

Zeus Components
100 Midland Avenue
Port Chester, NY 10573
(914) 937-7400
TWX 710-567-1248
FAX (914) 937-2553

NORTH CAROLINA
Dixie Electronics
2220 S. Tryon Street
Charlotte, NC 28234
(704) 377-4348

Dixie Electronics
1021 R. Burke St.
Winston-Salem, NC 27102
(919) 724-5961

Hammond Electronics
2923 Pacific Avenue
Greensboro, NC 27406
(919) 275-6391
TWX 628-94645

Pioneer Technologies Group
9801 A Southern Pine Blvd.
Charlotte, NC 28210
(704) 527-8188
TWX 810-621-0366

Quality Components, S.E.
2940-15 Trawick Road
Raleigh, NC 27604
(919) 876-7767

Schweber Electronics
5285 North Blvd.
Raleigh, NC 27604
(919) 876-0000
TWX 510-928-0531

NORTH DAKOTA
See Minnesota

OHIO
Kierulff Electronics
476 Windsor Park Drive
Dayton, OH 45459
(513) 439-0045

U.S. AND CANADIAN DISTRIBUTORS

OHIO (cont.)
Marshall Industries
6212 Executive Blvd.
Dayton, OH 45424
(513) 236-8088

Marshall Industries
59058 Harper Road
Solon, OH 44139
(216) 248-1788

Pioneer-Standard
4800 East 131st Street
Cleveland, OH 44105
(216) 587-3600
TWX 810-421-0011

Pioneer-Standard
4433 Interpoint Blvd.
Dayton, OH 45424
(513) 236-9900
TWX 810-459-1622

Schweber Electronics
23880 Commerce Park Rd.
Beachwood, OH 44122
(216) 464-2970
TWX 810-427-9441

Schweber Electronics
7865 Paragon Road
Suite 210
Dayton, OH 45459
(513) 439-1800

Zeus (Televox)
2593 Lance Drive
Dayton, OH 45409
(513) 294-4499
TWX 75-9251
FAX (513) 294-6620

OKLAHOMA
Quality Components
9934 East 21st South
Tulsa, OK 74129
(918) 664-8812
TWX 910-860-5459
629-28599

Schweber Electronics
4815 South Sheridan
Fountain Plaza, Suite 109
Tulsa, OK 74145
(918) 622-8000

OREGON
Almac Electronics Corp.
1885 N.W. 169th Place
Beaverton, OR 98006
(503) 629-8090
FAX (503) 645-0611
TWX 910-467-8743

Kierulff Electronics
14273 N.W. Science Park Drive
Portland, OR 97229
(503) 641-9150

Marshall Industries
8333 S.W. Cirrus Dr.
Beaverton, OR 97005
(503) 644-5050

PENNSYLVANIA
Almo Electronics, Inc.
9815 Roosevelt Blvd.
Philadelphia, PA 19114
(215) 698-4063
TLX 476-1218
FAX (215) 969-6768

Pioneer-Standard
259 Kappa Drive
Pittsburgh, PA 15238
(412) 782-2300
TWX 710- 795-3122

Pioneer Technologies Group
261 Gibraltar Road
Horsham, PA 19044
(215) 674-4000
TWX 510-665-6778

Schweber Electronics
900 Business Center Dr.
Horsham, PA 19044
(215) 441-0600
TWX 510-665-6540

Schweber Electronics
1000 R.I.D.C. Plaza
Suite 203
Pittsburgh, PA 15238
(412) 782-1600
TWX 810-427-9441

RHODE ISLAND
See Massachusetts
New York

SOUTH CAROLINA
Dixie Electronics
1900 Barnwell Street
Columbia, SC 29202
(803) 779-5332
TLX 810-666-2620
FAX (803) 765-9276

Dixie Electronics
4909 Pelham Rd.
Greenville, SC 29606
(803) 297-1435

Dixie Electronics
#6 Pepperhill Square
7525 Brandywine Road
N. Charleston, SC 29410
(803) 552-2671

SOUTH DAKOTA
See Minnesota

TENNESSEE
Dixie Electronics
Box 8215 Suncrest Drive
Gray, TN 37615
(615) 477-3838

Dixie Electronics
6408 Clinton Highway
Knoxville, TN 27912
(615) 938-4131

TEXAS
Kierulff Electronics
3007 Longhorn Blvd.
Austin, TX 78759

Kierulff Electronics
9610 Skillman Ave.
Dallas, TX 75243
(214) 343-2400

Marshall Industries
2045 Chenault St.
Carrollton, TX 75006
(214) 233-5200
FAX (214) 770-0675

Pioneer-Standard
13710 Omega Road
Dallas, TX 75234
(214) 386-7300
TWX 910-860-5563

Pioneer-Standard
9901 Burnet Road
Austin, TX 78758
(512) 835-4000
TWX 910-874-1323

Pioneer-Standard
5853 Point West Drive
Houston, TX 77036
(713) 988-5555
TWX 910-881-1606

Quality Components
4257 Kellway Circle
Addison, TX 75001
(214) 733-4300
TWX 910-860-5459

Quality Components
1005 Industrial Blvd.
Sugarland, TX 77478
(713) 240-2255
TWX 629 27026

Quality Components
2120 M. Braker Lane
Austin, TX 78758
(512) 835-0220
TWX 324930

Schweber Electronics
4202 Beltway Drive
Dallas, TX 75234
(214) 661-5010
TWX 910-860-5493

Schweber Electronics
6300 La Calma Drive
Suite 240
Austin, TX 78752
(512) 458-8253
TWX 910-874-2045

Schweber Electronics
10625 Richmond, Suite 100
Houston, TX 77042
(713) 784-3600
TWX 910-881-4836

Zeus Components
1800 North Glenville
Suite 120
Richardson, TX 75081
(214) 783-7010
TWX 910-867-9422
FAX (214) 234-4385

UTAH
Integrated Electronics Corp.
101 N. 700 West
N. Salt Lake City, UT 84054
(801) 298-1869

Kierulff Electronics
1846 Parkway Blvd.
Salt Lake City, UT 84119
(801) 973-6913

Marshall Industries
3501 South Main St.
Salt Lake City, UT 84115
(801) 261-0901

VERMONT
See New York

VIRGINIA
See Maryland

WASHINGTON
Almac Electronics Corp.
14360 S.E. Eastgate Way
Bellevue, WA 98007
(206) 643-9992
TWX 910-444-2067
FAX (206) 746-7425

Almac Electronics Corp.
East 10905 Montgomery
Spokane, WA 99206
(509) 924-9500
TWX 510-773-1855
FAX (509) 928-6096

Kierulff Electronics
19450 68th Ave.
South Kent, WA 98032
(206) 575-4420

Marshall Industries
14102 N.E. 21st St.
Bellevue, WA 98007
(206) 747-9100

WASHINGTON D.C.
See Maryland

WEST VIRGINIA
See Ohio
Pennsylvania
Maryland

WISCONSIN
Kierulff Electronics
2238-E West Bluemound Road
Waukesha, WI 53186
(414) 784-8160

Marshall Industries
235 North Executive Dr.
#305
Brookfield, WI 53005
(414) 797-8400

Schweber Electronics
3050 South Calhoun Rd.
New Berlin, WI 53151
(414) 784-9020

WYOMING
See Oregon
Washington

CANADA
R.A.E. Industrial
3455 Gardner Court
Burnaby, B.C.
(604) 291-8866
TWX 610-929-3065

R.A.E. Industrial
11680 170th Street
Edmonton, Alberta
T5S 1J7
(403) 451-4001
TWX 037-2653

Zentronics
8 Tilbury Court
Brampton, Ontario
L6T 3T4
(416) 451-9600
TWX 06-97678
FAX (416) 451-8320

U.S. AND CANADIAN DISTRIBUTORS

CANADA (cont.)

Zentronics
3300-14 Ave., NE Bay #1
Calgary, Alberta
T2A 6J4
(403) 272-1021

Zentronics
155 Colonnade, S. #17/18
Nepean, Ontario
K2E 7K1
(613) 226-8840
TWX 06-97698

Zentronics
11400 Bridgeport Rd. #108
Richmond, B.C.
V6X 1T2
(604) 273-5575
TWX 04-355844

Zentronics
817 McCaffrey Street
St. Laurent, Quebec
H4T 1N3
(514) 737-9700
TWX 05-824826

Zentronics
564 Weber Street, N. #10
Waterloo, Ontario
N21 5C6
(519) 884-5700
TWX 06-97678

Zentronics
590 Berry Street
Winnipeg, Manitoba
R3H 0S1
(204) 775-8661
TWX 06-97678

Future Electronics
3220 5th Avenue, N.E.
Calgary, Alberta
T2A 5N1
(403) 235-5325

Future Electronics
82 St. Regis Crescent N.
Downsview, Ontario
M3J 1Z3
(416) 638-4771
TWX 610-491-1470
FAX (416) 638-2936

Future Electronics
5312 Calgary Trail South
Edmonton, Alberta
T6H 4J8
(403) 438-2858

Future Electronics
Hymus Blvd.
Pointe Claire
Montreal, Quebec
H9R 5C7
(514) 694-7710
TWX 610-421-3251 or
610-421-3500
FAX (514) 695-3707 or
(514) 694-0062

Future Electronics
Baxter Center
1050 Baxter Road
Ottawa, Ontario
K2C 3P2
(613) 820-8313
TWX 610-563-1697
FAX (613) 820-3271

Future Electronics
1695 Boundary Road
Vancouver, B.C.
B5K 4X7
(604) 294-1166
TLX 04354744
FAX (604) 294-1206

Future Electronics
444 Sharon Bay
Winnipeg, Manitoba
R2G 0H7
(604) 294-1166 (Vancouver)

CHAPTER 1 - 6800 MICROPROCESSORS

6800 MICROPROCESSORS SELECTION GUIDE

Part number	Description	Technology	Alt source	CLK freq. (MHz)	Page
EF6802 EF68A02 EF68B02	8-bit MPU - 128 bytes of RAM - On-chip oscillator Expandable up to 64 Kbytes - 72 instructions 7 addressing modes - 6800 compatible	NMOS	MC6802 MC68A02 MC68B02	1 1.5 2	1-3
EF6803 EF6803-1 EF68A03 EF68B03	8-bit MPU - 128 bytes of RAM - Multiplexed address and data bus - 16-bit address bus - 8 × 8 multiply - Serial communication interface - 16-bit timer - 6800 compatible	HMOS	MC6803 MC6803-1 MC68A03 MC68B03	1 1.25 1.5 2	1-27
EF6803U4 EF6803U4-1 EF68A03U4	8-bit MPU - 192 bytes of RAM - Multiplexed address and data bus - 16-bit address bus - 8 × 8 multiply - Serial communication interface - 16-bits enhanced timer - 6800 compatible	HMOS	MC6803U4 MC6803U4-1 MC68A03U4	1 1.25 1.5	1-67
EF6809 EF68A09 EF68B09	High performance 8-bit MPU with on-chip clock - 64 Kbytes addressing space - Internal 16-bit structure - 59 instruction types - 10 addressing modes - 6800 compatible	HMOS	MC6809 MC68A09 MC68B09	1 1.5 2	1-111
EF6809E EF68A09E EF68B09E	External clock version of EF6809	HMOS	MC6809E MC68A09E MC68B09E	1 1.5 2	1-151

EF6802 • EF6808

MICROPROCESSOR WITH CLOCK AND OPTIONAL RAM

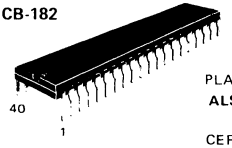
The EF6802 is a monolithic 8-bit microprocessor that contains all the registers and accumulators of the present EF6800 plus an internal clock oscillator and driver on the same chip. In addition, the EF6802 has 128 bytes of on-board RAM located at hex addresses $0000 to $007F. The first 32 bytes of RAM, at hex addresses $0000 to $001F, may be retained in a low power mode by utilizing V_{CC} standby ; thus, facilitating memory retention during a power-down situation.
The EF6802 is completely software compatible with the EF6800 as well as the entire EF6800 family of parts. Hence, the EF6802 is expandable to 64K words.
The EF6808 is identical to the EF6802 without on-board RAM.

- On-Chip Clock Circuit
- 128 x 8 Bit On-Chip RAM
- 32 Bytes of RAM are Retainable
- Software-Compatible with the EF6800
- Expandable to 64K Words
- Standard TTL-Compatible Inputs and Outputs
- 8-Bit Word Size
- 16-Bit Memory Addressing
- Interrupt Capability
- Three available versions : EF6802/08 (1.0 MHz), EF68A02/08 (1.5 MHz), EF68B02/08 (2.0 MHz).

MOS
(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)
MICROPROCESSOR
WITH CLOCK AND OPTIONAL RAM

CASES

CB-182

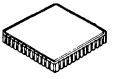


P SUFFIX
PLASTIC PACKAGE
ALSO AVAILABLE
C SUFFIX
CERAMIC PACKAGE

CB-521

FN SUFFIX
PLCC 44

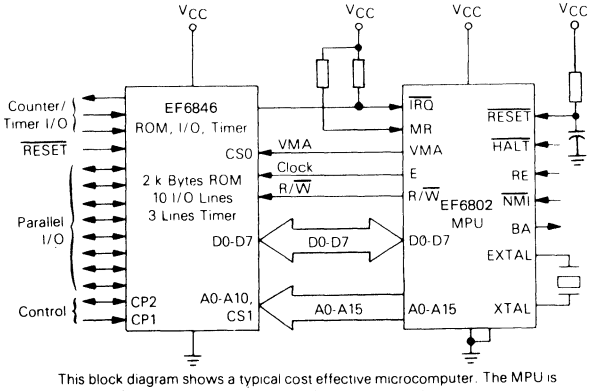
CB-708



E SUFFIX
LCCC 44

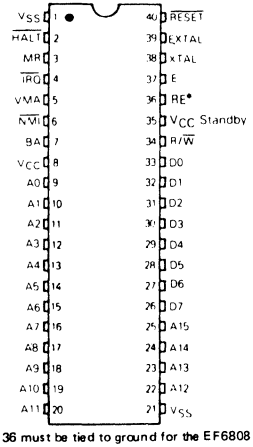
Hi-Rel versions available - See chapter 9

TYPICAL MICROCOMPUTER



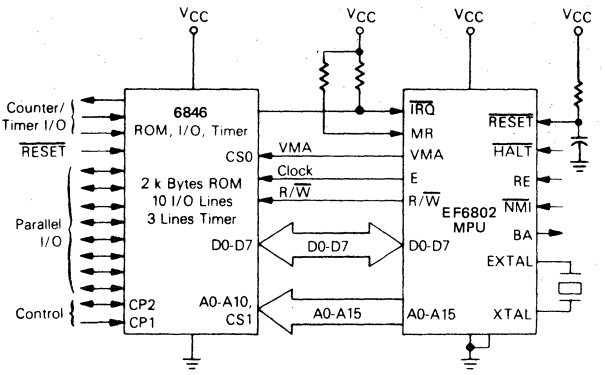
This block diagram shows a typical cost effective microcomputer. The MPU is the center of the microcoputer system and is shown in a minimum system interfacing with a ROM combination chip. It is not intended that this system be limited to this function but that it be expandable with other parts in the 6800 Microcomputer family.

PIN ASSIGNMENT



*Pin 36 must be tied to ground for the EF6808

TYPICAL MICROCOMPUTER



This block diagram shows a typical cost effective microcomputer. The MPU is the center of the microcomputer system and is shown in a minimum system interfacing with a ROM combination chip. It is not intended that this system be limited to this function but that it be expandable with other parts in the 6800 Microcomputer family.

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range EF6802, EF680A02, EF680B02 EF6802, EF68A02, EF68B02 : V suffix EF6802, EF68A02 : M suffix EF6808, EF68A08, EF68B08	T_A	 0 to +70 −40 to +85 −55 to +125 0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This input contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Average Thermal Resistance (Junction to Ambient) Plastic Ceramic PLCC	θ_{JA}	 100 50 100	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 0 to 70°C, unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, EXTAL RESET	V_{IH}	V_{SS} + 2.0 V_{SS} + 4.0	– –	V_{CC} V_{CC}	V
Input Low Voltage	Logic, EXTAL, RESET	V_{IL}	V_{SS} – 0.3	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V, V_{CC} = max)	Logic	I_{in}	–	1.0	2.5	μA
Output High Voltage (I_{Load} = – 205 μA, V_{CC} = min) (I_{Load} = – 145 μA, V_{CC} = min) (I_{Load} = – 100 μA, V_{CC} = min)	 D0-D7 A0-A15, R/W, VMA, E BA	V_{OH}	 V_{SS} + 2.4 V_{SS} + 2.4 V_{SS} + 2.4	 – – –	 – – –	V
Output Low Voltage (I_{Load} = 1.6 mA, V_{CC} = min)		V_{OL}	–	–	V_{SS} + 0.4	V
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	–	0.750	1.0	W
V_{CC} Standby	Power Down Power Up	V_{SBB} V_{SB}	4.0 4.75	– –	5.25 5.25	V
Standby Current		I_{SBB}	–	–	8.0	mA
Capacitance # (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	 D0-D7 Logic Inputs, EXTAL	C_{in}	 – –	 10 6.5	 12.5 10	pF
	A0-A15, R/W, VMA	C_{out}	–	–	12	pF

*In power-down mode, maximum power dissipation is less than 42 mW.
#Capacitances are periodically sampled rather than 100% tested.

CONTROL TIMING (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = T_L to T_H, unless otherwise noted)

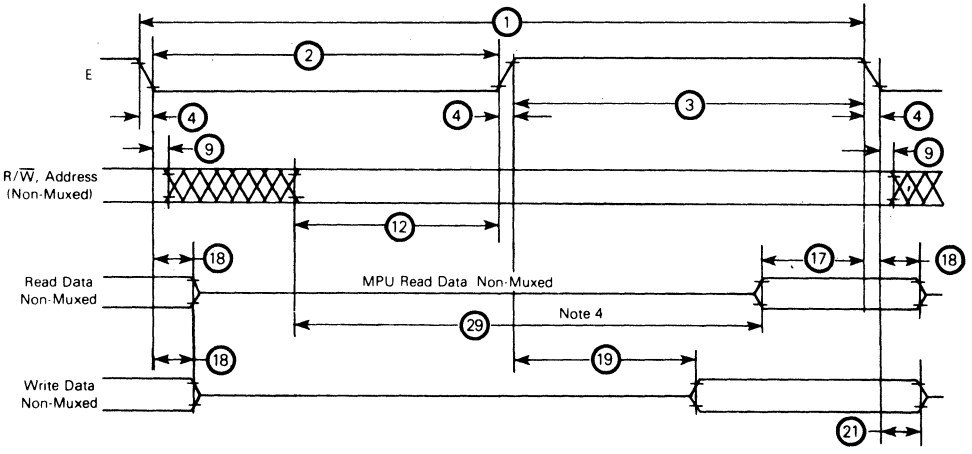
Characteristics	Symbol	EF6802 EF6808		EF68A02 EF68A08		EF68B02 EF68B08		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_O	0.1	1.0	0.1	1.5	0.1	2.0	MHz
Crystal Frequency	f_{XTAL}	1.0	4.0	1.0	6.0	1.0	8.0	MHz
External Oscillator Frequency	$4xf_O$	0.4	4.0	0.4	6.0	0.4	8.0	MHz
Crystal Oscillator Start Up Time	t_{rc}	100	–	100	–	100	–	ms
Processor Controls (HALT, MR, RE, RESET, IRQ NMI) Processor Control Setup Time Processor Control Rise and Fall Time (Does Not Apply to RESET)	 t_{PCS} t_{PCr}, t_{PCf}	 200 –	 – 100	 140 –	 – 100	 110 –	 – 100	 ns ns

BUS TIMING CHARACTERISTICS

Ident. Number	Characteristic	Symbol	EF6802 EF6808		EF68A02 EF68A08		EF68B02 EF68B08		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.667	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	450	5000	280	5000	210	5000	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9700	220	9700	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	25	ns
9	Address Hold Time*	t_{AH}	20	–	20	–	20	–	ns
12	Non-Muxed Address Valid Time to E (See Note 5)	t_{AV1} t_{AV2}	160 –	– 270	100 –	– –	50 –	– –	ns
17	Read Data Setup Time	t_{DSR}	100	–	70	–	60	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	170	–	160	ns
21	Write Data Hold Time*	t_{DHW}	30	–	20	–	20	–	ns
29	Usable Access Time (See Note 4)	t_{ACC}	535	–	335	–	235	–	ns

* Address and data hold times are periodically tested rather than 100% tested.

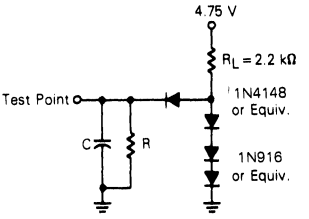
FIGURE 2 – BUS TIMING



NOTES:

1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise noted.
3. All electricals shown for the EF6802 apply to the EF6808, unless otherwise noted.
4. Usable access time is computed by: 12 + 3 + 4 – 17.
5. If programs are not executed from on-board RAM, TAV1 applies. If programs are to be stored and executed from on-board RAM, TAV2 applies. For normal data storage in the on-board RAM, this extended delay does not apply. Programs cannot be executed from on-board RAM when using A and B parts (EF68A02, EF68A08, EF68B02, EF68B08). On-board RAM can be used for data storage with all parts.
6. All electrical and control characteristics are referenced from: $T_L = 0°C$ minimum and $T_H = 70°C$ maximum.

FIGURE 3 — BUS TIMING TEST LOAD



C = 130 pF for D0-D7, E
= 90 pF for A0-A15, R/$\overline{W}$, and VMA
= 30 pF for BA
R = 11.7 kΩ for D0-D7, E
= 16.5 kΩ for A0-A15, R/$\overline{W}$, and VMA
= 24 kΩ for BA

FIGURE 4 — TYPICAL DATA BUS OUTPUT DELAY versus CAPACITIVE LOADING

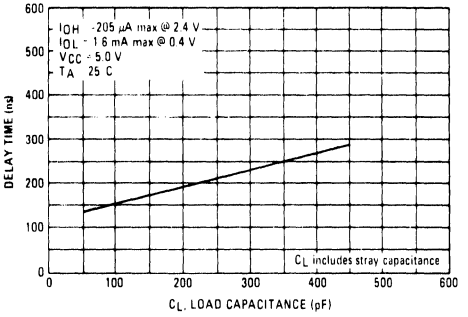


FIGURE 5 — TYPICAL READ/WRITE, VMA AND ADDRESS OUTPUT DELAY versus CAPACITIVE LOADING

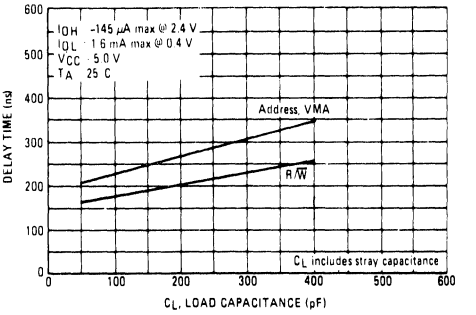
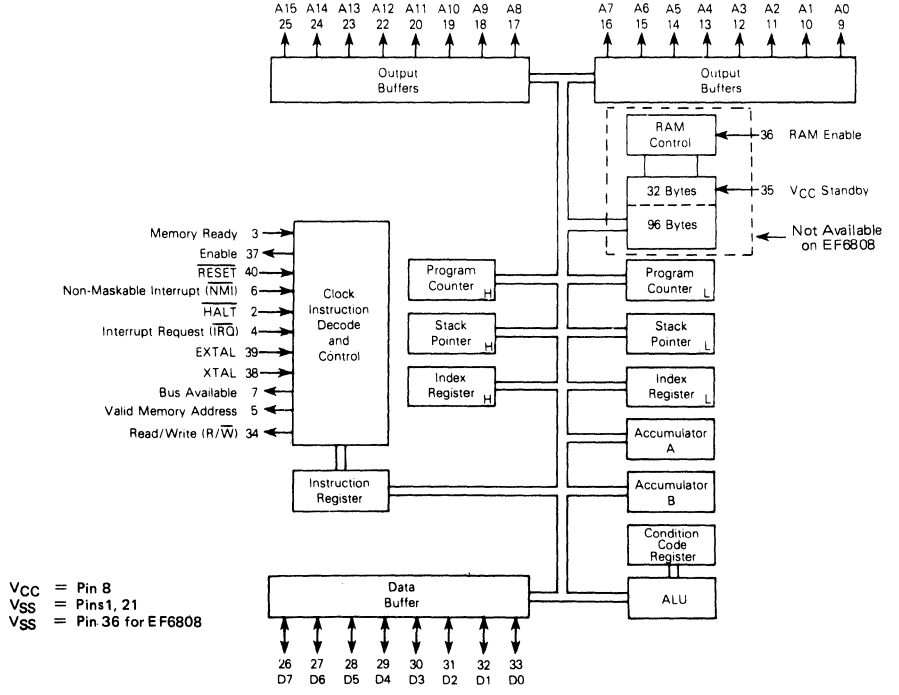


FIGURE 6 — EXPANDED BLOCK DIAGRAM



V_{CC} = Pin 8
V_{SS} = Pins 1, 21
V_{SS} = Pin 36 for EF6808

MPU REGISTERS

A general block diagram of the EF6802 is shown in Figure 6. As shown, the number and configuration of the registers are the same as for the EF6800. The 128x8-bit RAM* has been added to the basic MPU. The first 32 bytes can be retained during power-up and power-down conditions via the RE signal.

The EF6808 is identical to the EF6802 except for on-board RAM. Since the EF6808 does not have on-board RAM pin 36 must be tied to ground allowing the processor to utilize up to 64K bytes of external memory.

The MPU has three 16-bit registers and three 8-bit registers available for use by the programmer (Figure 7).

PROGRAM COUNTER

The program counter is a two byte (16-bit) register that points to the current program address.

STACK POINTER

The stack pointer is a two byte register that contains the address of the next available location in an external push-down/pop-up stack. This stack is normally a random access read/write memory that may have any location (address) that is convenient. In those applications that require storage of information in the stack when power is lost, the stack must be non-volatile.

INDEX REGISTER

The index register is a two byte register that is used to store data or a 16-bit memory address for the indexed mode of memory addressing.

ACCUMULATORS

The MPU contains two 8-bit accumulators that are used to hold operands and results from an arithmetic logic unit (ALU).

CONDITION CODE REGISTER

The condition code register indicates the results of an Arithmetic Logic Unit operation: Negative (N), Zero (Z), Overflow (V), Carry from bit 7 (C), and Half Carry from bit 3 (H). These bits of the Condition Code Register are used as testable conditions for the conditional branch instructions. Bit 4 is the interrupt mask bit (I). The unused bits of the Condition Code Register (b6 and b7) are ones.

Figure 8 shows the order of saving the microprocessor status within the stack.

*If programs are not executed from on-board RAM, TAV1 applies. If programs are to be stored and executed from on-board RAM, TAV2 applies. For normal data storage in the on-board RAM, this extended delay does not apply. Programs cannot be executed from on-board RAM when using A and B parts (EF68A02, EF68A08, EF68B02, and EF68B08). On-board RAM can be used for data storage with all parts.

FIGURE 7 — PROGRAMMING MODEL OF THE MICROPROCESSING UNIT

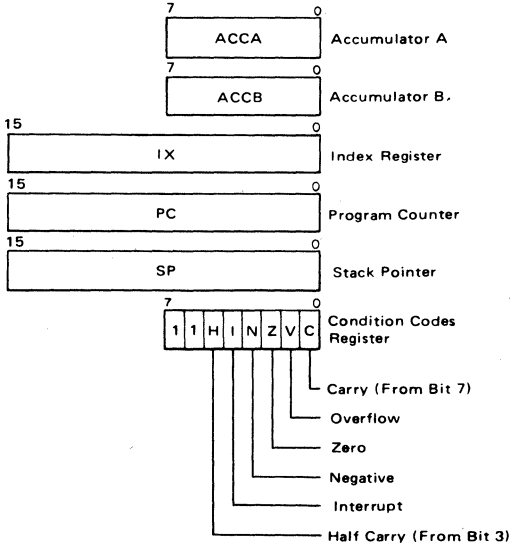
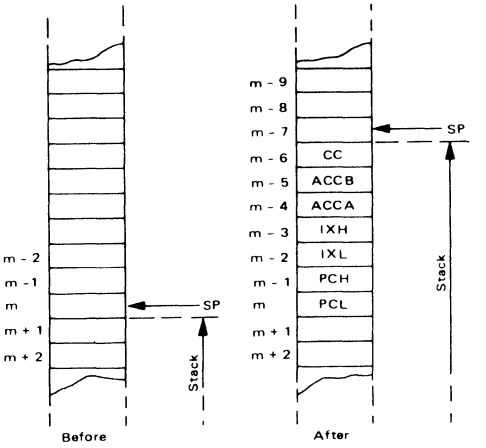


FIGURE 8 — SAVING THE STATUS OF THE MICROPROCESSOR IN THE STACK

SP = Stack Pointer
CC = Condition Codes (Also called the Processor Status Byte)
ACCB = Accumulator B
ACCA = Accumulator A
IXH = Index Register, Higher Order 8 Bits
IXL = Index Register, Lower Order 8 Bits
PCH = Program Counter, Higher Order 8 Bits
PCL = Program Counter, Lower Order 8 Bits



MPU SIGNAL DESCRIPTION

Proper operation of the MPU requires that certain control and timing signals be provided to accomplish specific functions and that other signal lines be monitored to determine the state of the processor. These control and timing signals are similar to those of the EF6800 except that TSC, DBE, ϕ1, ϕ2 input, and two unused pins have been eliminated, and the following signal and timing lines have been added:

RAM Enable (RE)
Crystal Connections EXTAL and XTAL
Memory Ready (MR)
V_{CC} Standby
Enable ϕ2 Output (E)

The following is a summary of the MPU signals:

ADDRESS BUS (A0-A15)

Sixteen pins are used for the address bus. The outputs are capable of driving one standard TTL load and 90 pF. These lines do not have three-state capability.

DATA BUS (D0-D7)

Eight pins are used for the data bus. It is bidirectional, transferring data to and from the memory and peripheral devices. It also has three-state output buffers capable of driving one standard TTL load and 130 pF.

Data bus will be in the output mode when the internal RAM is accessed and RE will be high. This prohibits external data entering the MPU. It should be noted that the internal RAM is fully decoded from $0000 to $007F. External RAM at $0000 to $007F must be disabled when internal RAM is accessed.

$\overline{\text{HALT}}$

When this input is in the low state, all activity in the machine will be halted. This input is level sensitive. In the $\overline{\text{HALT}}$ mode, the machine will stop at the end of an instruction, bus available will be at a high state, valid memory address will be at a low state. The address bus will display the address of the next instruction.

To ensure single instruction operation, transition of the $\overline{\text{HALT}}$ line must occur t_{PCS} before the falling edge of E and the $\overline{\text{HALT}}$ line must go high for one clock cycle.

$\overline{\text{HALT}}$ should be tied high if not used. This is good engineering design practice in general and necessary to ensure proper operation of the part.

READ/WRITE (R/$\overline{\text{W}}$)

This TTL-compatible output signals the peripherals and memory devices whether the MPU is in a read (high) or write (low) state. The normal standby state of this signal is read (high). When the processor is halted, it will be in the read state. This output is capable of driving one standard TTL load and 90 pF.

VALID MEMORY ADDRESS (VMA)

This output indicates to peripheral devices that there is a valid address on the address bus. In normal operation, this signal should be utilized for enabling peripheral interfaces such as the PIA and ACIA. This signal is not three-state. One standard TTL load and 90 pF may be directly driven by this active high signal.

BUS AVAILABLE (BA) — The bus available signal will normally be in the low state; when activated, it will go to the high state indicating that the microprocessor has stopped and that the address bus is available (but not in a three-state condition). This will occur if the $\overline{\text{HALT}}$ line is in the low state or the processor is in the WAIT state as a result of the execution of a WAIT instruction. At such time, all three-state output drivers will go to their off-state and other outputs to their normally inactive level. The processor is removed from the

WAIT state by the occurrence of a maskable (mask bit I = 0) or nonmaskable interrupt. This output is capable of driving one standard TTL load and 30 pF.

INTERRUPT REQUEST ($\overline{IRQ}$)

A low level on this input requests that an interrupt sequence be generated within the machine. The processor will wait until it completes the current instruction that is being excuted before it recognizes the request. At that time, if the interrupt mask bit in the condition code register is not set, the machine will begin an interrupt sequence. The index register, program counter, accumulators, and condition code register are stored away on the stack. Next the MPU will respond to the interrupt request by setting the interrupt mask bit high so that no further interrupts may occur. At the end of the cycle, a 16-bit vectoring address which is located in memory locations $FFF8 and $FFF9 is loaded which causes the MPU to branch to an interrupt routine in memory.

The $\overline{HALT}$ line must be in the high state for interrupts to be serviced. Interrupts will be latched internally while $\overline{HALT}$ is low.

A nominal 3 kΩ pullup resistor to V_{CC} should be used for wire-OR and optimum control of interrupts. $\overline{IRQ}$ may be tied directly to V_{CC} if not used.

$\overline{RESET}$

This input is used to reset and start the MPU from a power-down condition, resulting from a power failure or an initial start-up of the processor. When this line is low, the MPU is inactive and the information in the registers will be lost. If a high level is detected on the input, this will signal the MPU to begin the restart sequence. This will start execution of a routine to initialize the processor from its reset condition. All the higher order address lines will be forced high. For the restart, the last two ($FFFE, $FFFF) locations in memory will be used to load the program that is addressed by the program counter. During the restart routine, the interrupt mask bit is set and must be reset before the MPU can be interrupted by $\overline{IRQ}$. Power-up and reset timing and power-down sequences are shown in Figures 9 and 10, respectively.

$\overline{RESET}$, when brought low, must be held low at least three clock cycles. This allows adequate time to respond internally to the reset. This is independent of the t_{rc} power-up reset that is required.

When $\overline{RESET}$ is released it *must* go through the low-to-high threshold without bouncing, oscillating, or otherwise causing an erroneous reset (less than three clock cycles). This may cause improper MPU operation until the next valid reset.

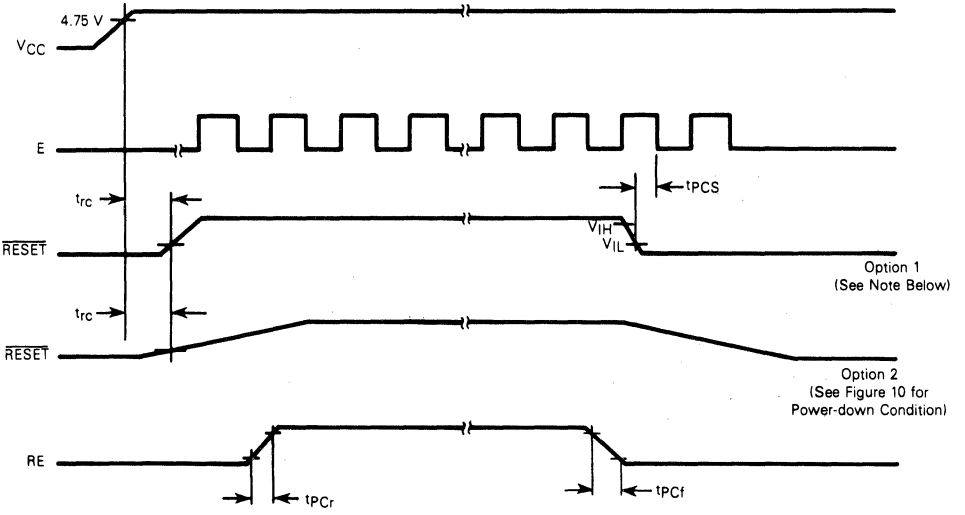
NON-MASKABLE INTERRUPT ($\overline{NMI}$)

A low-going edge on this input requests that a nonmaskable interrupt sequence be generated within the processor. As with the interrupt request signal, the processor will complete the current instruction that is being executed before it recognizes the $\overline{NMI}$ signal. The interrupt mask bit in the condition code register has no effect on $\overline{NMI}$.

The index register, program counter, accumulators, and condition code registers are stored away on the stack. At the end of the cycle, a 16-bit vectoring address which is located in memory locations $FFFC and $FFFD is loaded causing the MPU to branch to an interrupt service routine in memory.

A nominal 3 kΩ pullup resistor to V_{CC} should be used for wire-OR and optimum control of interrupts. $\overline{NMI}$ may be tied

FIGURE 9 — POWER-UP AND RESET TIMING



NOTE: If option 1 is chosen, $\overline{RESET}$ and RE pins can be tied together.

directly to V_{CC} if not used.

Inputs $\overline{IRQ}$ and $\overline{NMI}$ are hardware interrupt lines that are sampled when E is high and will start the interrupt routine on a low E following the completion of an instruction.

Figure 11 is a flowchart describing the major decision paths and interrupt vectors of the microprocessor. Table 1 gives the memory map for interrupt vectors.

TABLE 1 — MEMORY MAP FOR INTERRUPT VECTORS

Vector		Description
MS	LS	
$FFFE	$FFFF	Restart
$FFFC	$FFFD	Non-Maskable Interrupt
$FFFA	$FFFB	Software Interrupt
$FFF8	$FFF9	Interrupt Request

FIGURE 10 — POWER-DOWN SEQUENCE

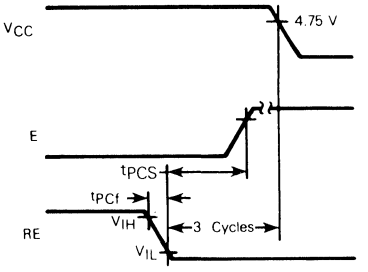


FIGURE 11 — MPU FLOWCHART

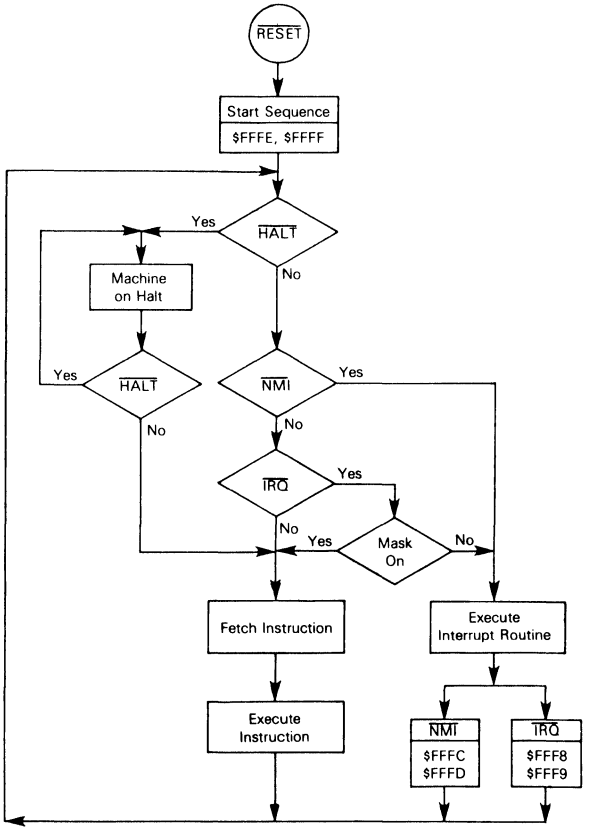
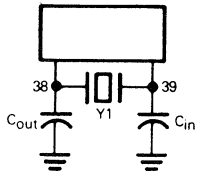
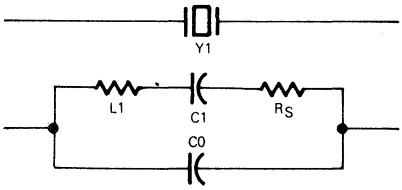


FIGURE 12 — CRYSTAL SPECIFICATIONS



Y1	C_{in}	C_{out}
3.58 MHz	27 pF	27 pF
. 4 MHz	27 pF	27 pF
6 MHz	20 pF	20 pF
8 MHz	18 pF	18 pF

Crystal Loading



Nominal Crystal Parameters*

	3.58 MHz	4.0 MHz	6.0 MHz	8.0 MHz
R_S	60 Ω	50 Ω	30-50 Ω	20-40 Ω
C0	3.5 pF	6.5 pF	4-6 pF	4-6 pF
C1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF
Q	> 40K	> 30K	> 20K	> 20K

*These are representative AT-cut parallel resonance crystal parameters only. Crystals of other types of cuts may also be used.

Figure 13 — SUGGESTED PC BOARD LAYOUT

Example of Board Design Using the Crystal Oscillator

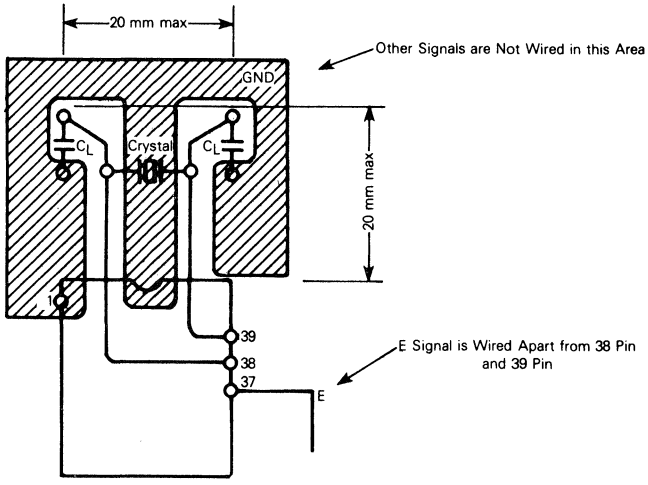


FIGURE 14 – MEMORY READY SYNCHRONIZATION

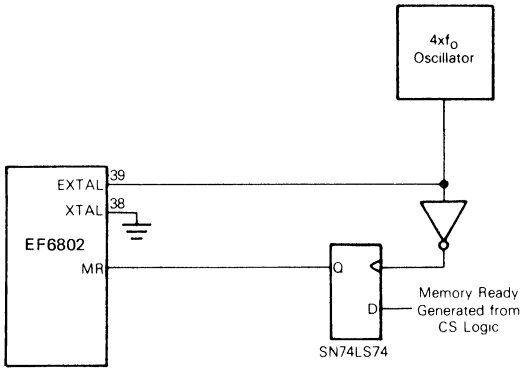
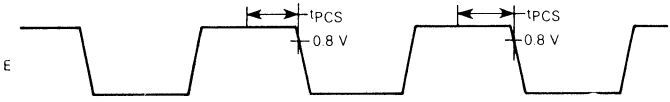


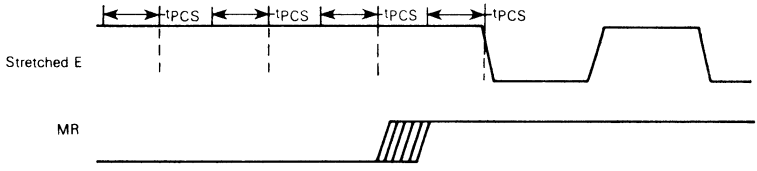
FIGURE 15 – MR NEGATIVE SETUP TIME REQUIREMENT

E Clock Stretch



The E clock will be stretched at end of E high of the cycle during which MR negative meets the t_{PCS} setup time. The t_{PCS} setup time is referenced to the fall of E. If the t_{PCS} setup time is not met, E will be stretched at the end of the next E-high ½ cycle. E will be stretched in integral multiples of ½ cycles.

Resuming E Clocking



The E clock will resume normal operation at the end of the ½ cycle during which MR assertion meets the t_{PCS} setup time. The t_{PCS} setup time is referenced to transitions of E were it not stretched. If t_{PCS} setup time is not met, E will fall at the second possible transition time after MR is asserted. There is no direct means of determining when the t_{PCS} references occur, unless the synchronizing circuit of Figure 14 is used.

RAM ENABLE (RE – EF6802 ONLY)

A TTL-compatible RAM enable input controls the on-chip RAM of the **EF6802**. When placed in the high state, the on-chip memory is enabled to respond to the MPU controls. In the low state, RAM is disabled. This pin may also be utilized to disable reading and writing the on-chip RAM during a power-down situation. RAM Enable must be low three cycles before V_{CC} goes below 4.75 V during power-down. RAM enable must be tied low on the **EF6808.** RE should be tied to the correct high or low state if not used.

EXTAL AND XTAL

These inputs are used for the internal oscillator that may be crystal controlled. These connections are for a parallel resonant fundamental crystal (see Figure 12). (AT-cut.) A divide-by-four circuit has been added so a 4 MHz crystal may be used in lieu of a 1 MHz crystal for a more cost-effective system. An example of the crystal circuit layout is shown in Figure 13. Pin 39 may be driven externally by a TTL input signal four times the required E clock frequency. Pin 38 is to be grounded.

An RC network is not directly usable as a frequency source on pins 38 and 39. An RC network type TTL or CMOS oscillator will work well as long as the TTL or CMOS output drives the on-chip oscillator.

LC networks are not recommended to be used in place of the crystal.

If an external clock is used, it may not be halted for more than $tPW\phi L$. The EF6802 and EF6808 are dynamic parts except for the internal RAM, and require the external clock to retain information.

MEMORY READY (MR)

MR is a TTL-compatible input signal controlling the stretching of E. Use of MR requires synchronization with the $4xf_o$ signal, as shown in Figure 14. When MR is high, E will be in normal operation. When MR is low, E will be stretched integral numbers of half periods, thus allowing interface to slow memories. Memory Ready timing is shown in Figure 15.

MR should be tied high (connected directly to V_{CC}) if not used. This is necessary to ensure proper operation of the part. A maximum stretch is t_{cyc}.

ENABLE (E)

This pin supplies the clock for the MPU and the rest of the system. This is a single-phase, TTL-compatible clock. This clock may be conditioned by a memory read signal. This is equivalent to $\phi 2$ on the **EF6800.** This output is capable of driving one standard TTL load and 130 pF.

V_{CC} STANDBY (EF6802 ONLY)

This pin supplies the dc voltage to the first 32 bytes of RAM as well as the RAM Enable (RE) control logic. Thus, retention of data in this portion of the RAM on a power-up, power-down, or standby condition is guaranteed. Maximum current drain at V_{SB} maximum is I_{SBB}.

MPU INSTRUCTION SET

The instruction set has 72 different instructions. Included are binary and decimal arithmetic, logical, shift, rotate, load, store, conditional or unconditional branch, interrupt and stack manipulation instructions (Tables 2 through 6). The instruction set is the same as that for the **EF6800.**

MPU ADDRESSING MODES

There are seven address modes that can be used by a programmer, with the addressing mode a function of both the type of instruction and the coding within the instruction. A summary of the addressing modes for a particular instruction can be found in Table 7 along with the associated instruction execution time that is given in machine cycles. With a bus frequency of 1 MHz, these times would be microseconds.

ACCUMULATOR (ACCX) ADDRESSING

In accumulator only addressing, either accumulator A or accumulator B is specified. These are one-byte instructions.

IMMEDIATE ADDRESSING

In immediate addressing, the operand is contained in the second byte of the instruction except LDS and LDX which have the operand in the second and third bytes of the instruction. The MPU addresses this location when it fetches the immediate instruction for execution. These are two- or three-byte instructions.

DIRECT ADDRESSING

In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in the machine, i.e., locations zero through 255. Enhanced execution times are achieved by storing data in these locations. In most configurations, it should be a random-access memory. These are two-byte instructions.

EXTENDED ADDRESSING

In extended addressing, the address contained in the second byte of the instruction is used as the higher eight bits of the address of the operand. The third byte of the instruction is used as the lower eight bits of the address for the operand. This is an absolute address in memory. These are three-byte instructions.

INDEXED ADDRESSING

In indexed addressing, the address contained in the second byte of the instruction is added to the index register's lowest eight bits in the MPU. The carry is then added to the higher order eight bits of the index register. This result is then used to address memory. The modified address is held in a temporary address register so there is no change to the index register. These are two-byte instructions.

IMPLIED ADDRESSING

In the implied addressing mode, the instruction gives the address (i.e., stack pointer, index register, etc.). These are one-byte instructions.

RELATIVE ADDRESSING

In relative addressing, the address contained in the second byte of the instruction is added to the program counter's lowest eight bits plus two. The carry or borrow is then added to the high eight bits. This allows the user to address data within a range of – 125 to + 129 bytes of the present instruction. These are two-byte instructions.

TABLE 2 – MICROPROCESSOR INSTRUCTION SET – ALPHABETIC SEQUENCE

ABA	Add Accumulators	CLR	Clear	PUL	Pull Data
ADC	Add with Carry	CLV	Clear Overflow	ROL	Rotate Left
ADD	Add	CMP	Compare	ROR	Rotate Right
AND	Logical And	COM	Complement	RTI	Return from Interrupt
ASL	Arithmetic Shift Left	CPX	Compare Index Register	RTS	Return from Subroutine
ASR	Arithmetic Shift Right	DAA	Decimal Adjust	SBA	Subtract Accumulators
BCC	Branch if Carry Clear	DEC	Decrement	SBC	Subtract with Carry
BCS	Branch if Carry Set	DES	Decrement Stack Pointer	SEC	Set Carry
BEQ	Branch if Equal to Zero	DEX	Decrement Index Register	SEI	Set Interrupt Mask
BGE	Branch if Greater or Equal Zero	EOR	Exclusive OR	SEV	Set Overflow
BGT	Branch if Greater than Zero	INC	Increment	STA	Store Accumulator
BHI	Branch if Higher	INS	Increment Stack Pointer	STS	Store Stack Register
BIT	Bit Test	INX	Increment Index Register	STX	Store Index Register
BLE	Branch if Less or Equal	JMP	Jump	SUB	Subtract
BLS	Branch if Lower or Same	JSR	Jump to Subroutine	SWI	Software Interrupt
BLT	Branch if Less than Zero	LDA	Load Accumulator	TAB	Transfer Accumulators
BMI	Branch if Minus	LDS	Load Stack Pointer	TAP	Transfer Accumulators to Condition Code Reg.
BNE	Branch if Not Equal to Zero	LDX	Load Index Register	TBA	Transfer Accumulators
BPL	Branch if Plus	LSR	Logical Shift Right	TPA	Transfer Condition Code Reg. to Accumulator
BRA	Branch Always	NEG	Negate	TST	Test
BSR	Branch to Subroutine	NOP	No Operation	TSX	Transfer Stack Pointer to Index Register
BVC	Branch if Overflow Clear	ORA	Inclusive OR Accumulator	TXS	Transfer Index Register to Stack Pointer
BVS	Branch if Overflow Set	PSH	Push Data	WAI	Wait for Interrupt
CBA	Compare Accumulators				
CLC	Clear Carry				
CLI	Clear Interrupt Mask				

TABLE 3 — ACCUMULATOR AND MEMORY INSTRUCTIONS

OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED =	DIRECT OP	DIRECT ~	DIRECT =	INDEX OP	INDEX ~	INDEX =	EXTND OP	EXTND ~	EXTND =	IMPLIED OP	IMPLIED ~	IMPLIED =	BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	5 H	4 I	3 N	2 Z	1 V	0 C
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	•	•	↕	↕	R	•
Clear	CLR							6F	7	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
Compare Acmltrs	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	7	2	73	6	3				M̄ → M	•	•	↕	↕	R	S
	COMA													43	2	1	Ā → A	•	•	↕	↕	R	S
	COMB													53	2	1	B̄ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	•	•	↕	↕	①	②
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	①	②
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	•	•	↕	↕	↕	③
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	•	•	↕	↕	④	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	④	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A⊕M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B⊕M → B	•	•	↕	↕	R	•
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	•	•	↕	↕	R	•
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, M_{SP} → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, M_{SP} → B	•	•	•	•	•	•
Rotate Left	ROL							69	7	2	79	6	3				M	•	•	↕	↕	⑥	↕
	ROLA													49	2	1	A	•	•	↕	↕	⑥	↕
	ROLB													59	2	1	B — C b7 b0	•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	7	2	76	6	3				M	•	•	↕	↕	⑥	↕
	RORA													46	2	1	A	•	•	↕	↕	⑥	↕
	RORB													56	2	1	B — C b7 b0	•	•	↕	↕	⑥	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	•	•	↕	↕	⑥	↕
	ASLA													48	2	1	A — 0	•	•	↕	↕	⑥	↕
	ASLB													58	2	1	B — C b7 b0	•	•	↕	↕	⑥	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	2	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	2	1	B — b7 b0 C	•	•	↕	↕	⑥	↕
Shift Right, Logic	LSR							64	7	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	2	1	A — 0	•	•	R	↕	⑥	↕
	LSRB													54	2	1	B — b7 b0 C	•	•	R	↕	⑥	↕
Store Acmltr	STAA				97	4	2	A7	6	2	B7	5	3				A → M	•	•	↕	↕	R	•
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subtract Acmltrs	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Transfer Acmltrs	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R
																		H	I	N	Z	V	C

LEGEND:

OP	Operation Code (Hexadecimal);	+	Boolean Inclusive OR;
~	Number of MPU Cycles;	⊕	Boolean Exclusive OR;
=	Number of Program Bytes;	M̄	Complement of M;
+	Arithmetic Plus;	→	Transfer Into;
−	Arithmetic Minus;	0	Bit = Zero;
·	Boolean AND;	00	Byte = Zero;
M_{SP}	Contents of memory location pointed to be Stack Pointer;		

Note – Accumulator addressing mode instructions are included in the column for IMPLIED addressing

CONDITION CODE SYMBOLS:

H	Half-carry from bit 3;
I	Interrupt mask
N	Negative (sign bit)
Z	Zero (byte)
V	Overflow, 2's complement
C	Carry from bit 7
R	Reset Always
S	Set Always
↕	Test and set if true, cleared otherwise
•	Not Affected

TABLE 4 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

		IMMED			DIRECT			INDEX			EXTND			IMPLIED				COND. CODE REG.					
																		5	4	3	2	1	0
POINTER OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BOOLEAN/ARITHMETIC OPERATION	H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$X_H - M, X_L - (M + 1)$	•	•	⑦	↕	⑧	•
Decrement Index Reg	DEX													09	4	1	$X - 1 \rightarrow X$	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	4	1	$SP - 1 \rightarrow SP$	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	$X + 1 \rightarrow X$	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	4	1	$SP + 1 \rightarrow SP$	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M \rightarrow X_H, (M + 1) \rightarrow X_L$	•	•	⑨	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \rightarrow SP_H, (M + 1) \rightarrow SP_L$	•	•	⑨	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \rightarrow M, X_L \rightarrow (M + 1)$	•	•	⑨	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \rightarrow M, SP_L \rightarrow (M + 1)$	•	•	⑨	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	$X - 1 \rightarrow SP$	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	$SP + 1 \rightarrow X$	•	•	•	•	•	•

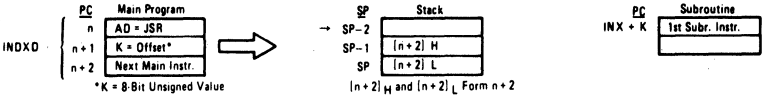
TABLE 5 — JUMP AND BRANCH INSTRUCTIONS

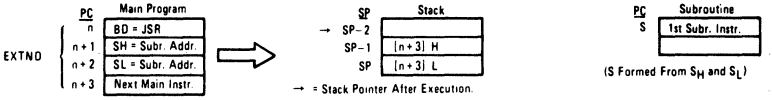
		RELATIVE			INDEX			EXTND			IMPLIED				COND. CODE REG.					
															5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BRANCH TEST	H	I	N	Z	V	C
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	4	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	4	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	8	2										}	•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3				} See Special Operations	•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3				} (Figure 16)	•	•	•	•	•	•
No Operation	NOP										01	2	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI										3B	10	1		———			⑩	———	
Return From Subroutine	RTS										39	5	1	}	•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1	} See Special Operations	•	•	•	•	•	•
Wait for Interrupt	WAI										3E	9	1	} (Figure 16)	•	⑪	•	•	•	•

FIGURE 16 — SPECIAL OPERATIONS

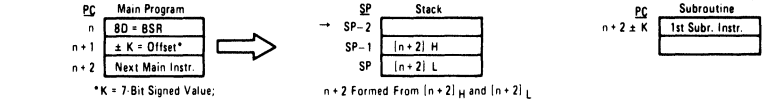
SPECIAL OPERATIONS

JSR, JUMP TO SUBROUTINE:



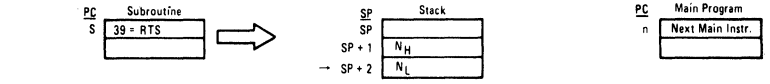


BSR, BRANCH TO SUBROUTINE:



JMP, JUMP:

RTS, RETURN FROM SUBROUTINE:



RTI, RETURN FROM INTERRUPT:

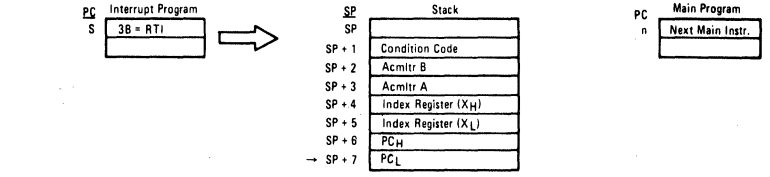


TABLE 6 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

		IMPLIED				COND. CODE REG.					
						5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	#	BOOLEAN OPERATION	H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Acmltr A → CCR	TAP	06	2	1	A → CCR	——— ⑫ ———					
CCR → Acmltr A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

CONDITION CODE REGISTER NOTES: (Bit set if test is true and cleared otherwise)

1 (Bit V) Test: Result = 10000000?
2 (Bit C) Test: Result ≠ 00000000?
3 (Bit C) Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set.)
4 (Bit V) Test: Operand = 10000000 prior to execution?
5 (Bit V) Test: Operand = 01111111 prior to execution?
6 (Bit V) Test: Set equal to result of N⊕C after shift has occurred.
7 (Bit N) Test: Sign bit of most significant (MS) byte = 1?
8 (Bit V) Test: 2's complement overflow from subtraction of MS bytes?
9 (Bit N) Test: Result less than zero? (Bit 15 = 1)
10 (All) Load Condition Code Register from Stack. (See Special Operations)
11 (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
12 (All) Set according to the contents of Accumulator A.

TABLE 7 — INSTRUCTION ADDRESSING MODES AND ASSOCIATED EXECUTION TIMES
(Times in Machine Cycle)

	(Dual Operand)	ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA		•	•	•	•	•	2	•
ADC	x	•	2	3	4	5	•	•
ADD	x	•	2	3	4	5	•	•
AND	x	•	2	3	4	5	•	•
ASL		2	•	•	6	7	•	•
ASR		2	•	•	6	7	•	•
BCC		•	•	•	•	•	•	4
BCS		•	•	•	•	•	•	4
BEA		•	•	•	•	•	•	4
BGE		•	•	•	•	•	•	4
BGT		•	•	•	•	•	•	4
BHI		•	•	•	•	•	•	4
BIT	x	•	2	3	4	5	•	•
BLE		•	•	•	•	•	•	4
BLS		•	•	•	•	•	•	4
BLT		•	•	•	•	•	•	4
BMI		•	•	•	•	•	•	4
BNE		•	•	•	•	•	•	4
BPL		•	•	•	•	•	•	4
BRA		•	•	•	•	•	•	4
BSR		•	•	•	•	•	•	8
BVC		•	•	•	•	•	•	4
BVS		•	•	•	•	•	•	4
CBA		•	•	•	•	•	2	•
CLC		•	•	•	•	•	2	•
CLI		•	•	•	•	•	2	•
CLR		2	•	•	6	7	•	•
CLV		•	•	•	•	•	2	•
CMP	x	•	2	3	4	5	•	•
COM		2	•	•	6	7	•	•
CPX		•	3	4	5	6	•	•
DAA		•	•	•	•	•	2	•
DEC		2	•	•	6	7	•	•
DES		•	•	•	•	•	4	•
DEX		•	•	•	•	•	4	•
EOR	x	•	2	3	4	5	•	•

	(Dual Operand)	ACCX	Immediate	Direct	Extended	Indexed	Implied
INC		2	•	•	6	7	•
INS		•	•	•	•	•	4
INX		•	•	•	•	•	4
JMP		•	•	•	3	4	•
JSR		•	•	•	9	8	•
LDA	x	•	2	3	4	5	•
LDS		•	3	4	5	6	•
LDX		•	3	4	5	6	•
LSR		2	•	•	6	7	•
NEG		2	•	•	6	7	•
NOP		•	•	•	•	•	2
ORA	x	•	2	3	4	5	•
PSH		•	•	•	•	•	4
PUL		•	•	•	•	•	4
ROL		2	•	•	6	7	•
ROR		2	•	•	6	7	•
RTI		•	•	•	•	•	10
RTS		•	•	•	•	•	5
SBA		•	•	•	•	•	2
SBC	x	•	2	3	4	5	•
SEC		•	•	•	•	•	2
SEI		•	•	•	•	•	2
SEV		•	•	•	•	•	2
STA	x	•	•	4	5	6	•
STS		•	•	5	6	7	•
STX		•	•	5	6	7	•
SUB	x	•	2	3	4	5	•
SWI		•	•	•	•	•	12
TAB		•	•	•	•	•	2
TAP		•	•	•	•	•	2
TBA		•	•	•	•	•	2
TPA		•	•	•	•	•	2
TST		2	•	•	6	7	•
TSX		•	•	•	•	•	4
TSX		•	•	•	•	•	4
WAI		•	•	•	•	•	9

NOTE: Interrupt time is 12 cycles from the end of the instruction being executed, except following a WAI instruction. Then it is 4 cycles.

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 8 provides a detailed description of the information present on the address bus, data bus, valid memory address line (VMA), and the read/write line (R/$\overline{W}$) during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing modes and number of cycles per instruction. (In general, instructions with the same addressing mode and number of cycles execute in the same manner; exceptions are indicated in the table.)

TABLE 8 — OPERATIONS SUMMARY

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data
CPX LDS LDX	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data (High Order Byte)
		3	1	Op Code Address + 2	1	Operand Data (Low Order Byte)
DIRECT						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	1	Address of Operand	1	Operand Data
CPX LDS LDX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	1	Address of Operand	1	Operand Data (High Order Byte)
		4	1	Operand Address + 1	1	Operand Data (Low Order Byte)
STA	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address
		3	0	Destination Address	1	Irrelevant Data (Note 1)
		4	1	Destination Address	0	Data from Accumulator
STS STX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	0	Address of Operand	1	Irrelevant Data (Note 1)
		4	1	Address of Operand	0	Register Data (High Order Byte)
		5	1	Address of Operand + 1	0	Register Data (Low Order Byte)
INDEXED						
JMP	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Operand Data
CPX LDS LDX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Operand Data (High Order Byte)
		6	1	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)

TABLE 8 — OPERATIONS SUMMARY (CONTINUED)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED (Continued)						
STA	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Current Operand Data
		6	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		7	1/0 (Note 3)	Index Register Plus Offset	0	New Operand Data (Note 3)
STS STX	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data (High Order Byte)
		7	1	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
JSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer — 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer — 2	1	Irrelevant Data (Note 1)
		7	0	Index Register	1	Irrelevant Data (Note 1)
		8	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
EXTENDED						
JMP	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	1	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data
CPX LDS LDX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data (High Order Byte)
		5	1	Address of Operand + 1	1	Operand Data (Low Order Byte)
STA A STA B	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	1	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	0	Operand Destination Address	1	Irrelevant Data (Note 1)
		5	1	Operand Destination Address	0	Data from Accumulator
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Current Operand Data
		5	0	Address of Operand	1	Irrelevant Data (Note 1)
		6	1/0 (Note 3)	Address of Operand	0	New Operand Data (Note 3)

TABLE 8 — OPERATIONS SUMMARY (CONTINUED)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED (Continued)						
STS STX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	0	Address of Operand	1	Irrelevant Data (Note 1)
		5	1	Address of Operand	0	Operand Data (High Order Byte)
		6	1	Address of Operand + 1	0	Operand Data (Low Order Byte)
JSR	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	1	Subroutine Starting Address	1	Op Code of Next Instruction
		5	1	Stack Pointer	0	Return Address (Low Order Byte)
		6	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		7	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		8	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		9	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
INHERENT						
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
DES DEX INS INX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Previous Register Contents	1	Irrelevant Data (Note 1)
		4	0	New Register Contents	1	Irrelevant Data (Note 1)
PSH	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Accumulator Data
		4	0	Stack Pointer – 1	1	Accumulator Data
PUL	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Operand Data from Stack
TSX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	0	New Index Register	1	Irrelevant Data (Note 1)
TXS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Index Register	1	Irrelevant Data
		4	0	New Stack Pointer	1	Irrelevant Data
RTS	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Address of Next Instruction (High Order Byte)
		5	1	Stack Pointer + 2	1	Address of Next Instruction (Low Order Byte)

TABLE 8 — OPERATIONS SUMMARY (CONCLUDED)

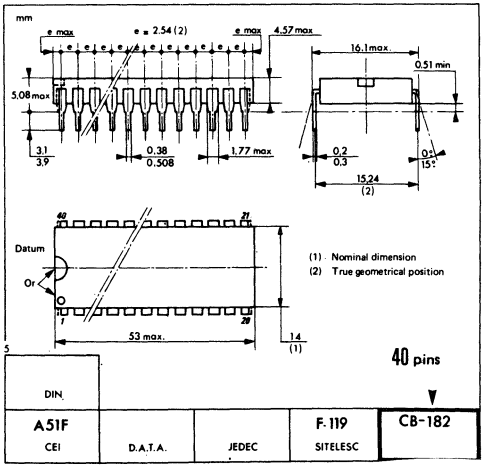
Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT (Continued)						
WAI	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Return Address (Low Order Byte)
		4	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	1	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	1	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	1	Stack Pointer – 4	0	Contents of Accumulator A
		8	1	Stack Pointer – 5	0	Contents of Accumulator B
		9	1	Stack Pointer – 6	1	Contents of Cond. Code Register
RTI	10	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Contents of Cond. Code Register from Stack
		5	1	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	1	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	1	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	1	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	1	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	1	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 1)
		3	1	Stack Pointer	0	Return Address (Low Order Byte)
		4	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	1	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	1	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	1	Stack Pointer – 4	0	Contents of Accumulator A
		8	1	Stack Pointer – 5	0	Contents of Accumulator B
		9	1	Stack Pointer – 6	0	Contents of Cond. Code Register
		10	0	Stack Pointer – 7	1	Irrelevant Data (Note 1)
		11	1	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	1	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE						
BCC BHI BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMI BVS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		4	0	Branch Address	1	Irrelevant Data (Note 1)
BSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		7	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		8	0	Subroutine Address (Note 4)	1	Irrelevant Data (Note 1)

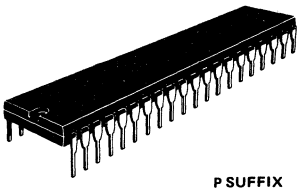
NOTES:

1. If device which is addressed during this cycle uses VMA, then the Data Bus will go to the high-impedance three-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the Data Bus.
2. Data is ignored by the MPU.
3. For TST, VMA = 0 and Operand data does not change.
4. MS Byte of Address Bus = MS Byte of Address of BSR instruction and LS Byte of Address Bus = LS Byte of Sub-Routine Address.

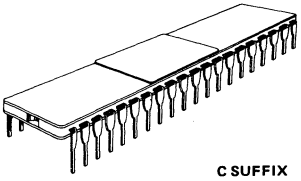
PHYSICAL DIMENSIONS

CB-182





P SUFFIX
PLASTIC PACKAGE



C SUFFIX
CERAMIC PACKAGE

ORDERING INFORMATION

EF68A02 | C | M | B/B

- EF68A02: Device
- C: Package
- M: Oper. temp.
- B/B: Screening level

The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

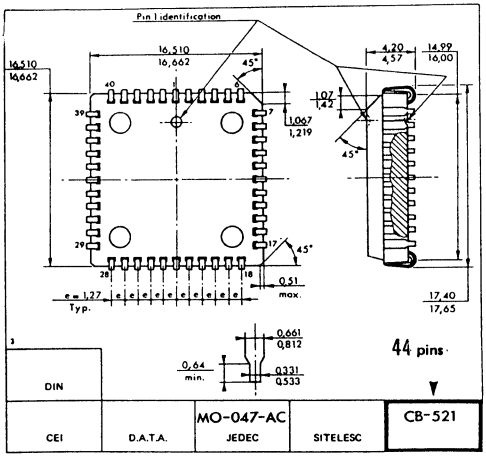
DEVICE	PACKAGE					OPER. TEMP			SCREENING LEVEL			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6802/08 (1.0 MHz)	●		●		●	●			●			
	●		●				●		●			
				●				●	●		●	●
EF68A02/A08 (1.5 MHz)	●		●			●			●			
	●		●				●		●			
	●			●				●	●		●	●
EF68B02/B08 (2.0 MHz)	●		●			●			●			
	●						●		●		●	

Examples : EF6802C, EF6802CV, EF6802EM, EF6802EM G/B

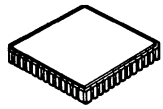
Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** −40°C to +85°C, **M :** −55°C to +125°C, ***** : may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.

CB-521

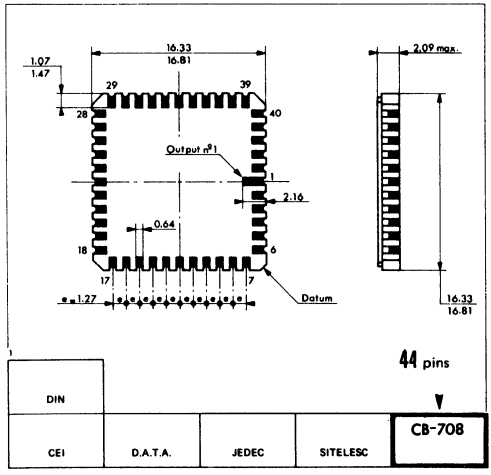
FN SUFFIX
PLCC 44



CB-708



E SUFFIX
LCCC 44



EF6801 • EF6803

MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

The EF6801 is an 8-bit single-chip microcomputer unit (MCU) which significantly enhances the capabilities of the 6800 family of parts. It includes an upgraded 6800 microprocessor unit (MPU) with upward-source and object-code compatibility. Execution times of key instructions have been improved and several new instructions have been added including an unsigned multiply. The MCU can function as a monolithic microcomputer or can be expanded to a 64K byte address space. It is TTL compatible and requires one + 5-volt power supply. On-chip resources include 2048 bytes of ROM, 128 bytes of RAM, a Serial Communications Interface (SCI), parallel I/O, and a three function Programmable Timer. The EF6803 can be considered as an EF6801 operating in Modes 2 or 3. EF6801 MCU Family features include :

- Enhanced EF6800 Instruction Set
- 8 x 8 Multiply Instruction
- Serial Communications Interface (SCI)
- Upward Source and Object Code Compatibility with the 6800
- 16-Bit Three-Function Programmable Timer
- Single-Chip or Expanded Operation to 64K Byte Address Space
- Bus Compatibility with the 6800 Family
- 2048 Bytes of ROM (EF6801)
- 128 Bytes of RAM
- 64 Bytes of RAM Retainable During Powerdown
- 29 Parallel I/O and Two Handshake Control Lines
- Internal Clock Generator with Divide-by-Four Output.
- Complete Development System Support on DEVICE®.
- −40°C to + 85°C Temperature range
- −40°C to + 105°C Temperature range

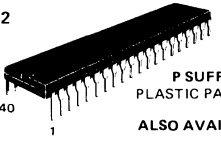
HMOS

(HIGH-DENSITY N-CHANNEL, SILICON-GATE)

MICROCOMPUTER/ MICROPROCESSOR

CASES

CB-182



P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

C SUFFIX
CERAMIC PACKAGE

CB-521

FN SUFFIX
PLCC 44

PIN ASSIGNMENT

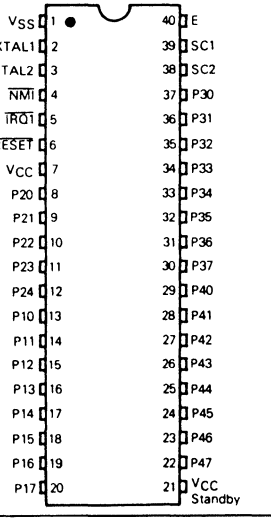
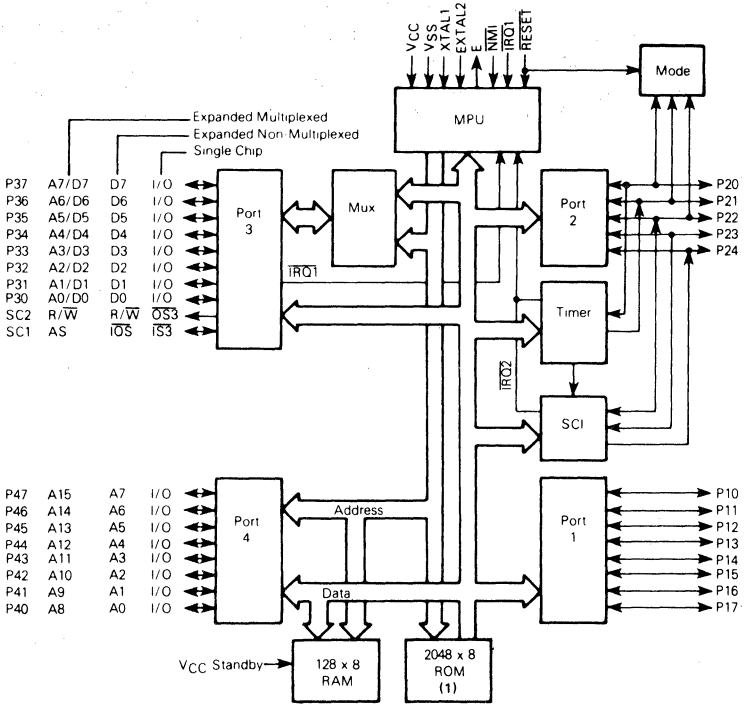


FIGURE 1 – 6801/6803 BLOCK DIAGRAM



(1) No functioning ROM in EF6803

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts – Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K + (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range EF6801/03, EF6801/03-1, EF68A01/03 EF68B01/03 EF6801/03, EF6801/03-1 : V suffix EF6801/03, EF6801/03-1 : A suffix	T_A	T_L to T_H 0 to 70 −40 to 85 −40 to 105	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$. Input protection is enhanced by connecting unused inputs to either V_{DD} or V_{SS}.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance Plastic PLCC	θ_{JA}	 50 100	 °C/W

CONTROL TIMING (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = 0 to 70°C)

Characteristic	Symbol	EF6801		EF6801-1		EF68A01		EF68B01		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	0.5	1.0	0.5	1.25	0.5	1.5	0.5	2.0	MHz
Crystal Frequency	f_{XTAL}	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
External Oscillator Frequency	$4f_o$	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
Crystal Oscillator Start Up Time	t_{rc}	–	100	–	100	–	100	–	100	ms
Processor Control Setup Time	t_{PCS}	200	–	170	–	140	–	110	–	ns

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H, unless otherwise noted)

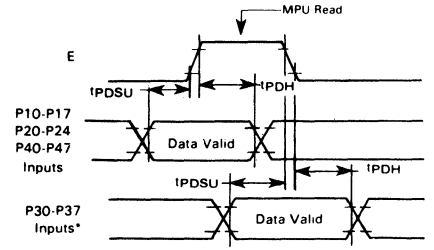
Characteristic		Symbol	EF6801/03 0°C to +70°C		EF6801/03 −40°C to +85°C		EF6801/03 −40°C to +105°C		Unit
			Min	Max	Min	Max	Min	Max	
Input High Voltage	$\overline{RESET}$	V_{IH}	$V_{SS}+4.0$	V_{CC}	$V_{SS}+4.0$	V_{CC}	$V_{SS}+4.0$	V_{CC}	V
	Other Inputs		$V_{SS}+2.0$	V_{CC}	$V_{SS}+2.2$	V_{CC}	$V_{SS}+2.2$	V_{CC}	
Input Low Voltage	All Inputs	V_{IL}	$V_{SS}-0.3$	$V_{SS}+0.8$	$V_{SS}-0.3$	$V_{SS}+0.8$	$V_{SS}-0.3$	$V_{SS}+0.8$	V
Input Load Current	Port 4	I_{in}	–	0.5	–	0.8	–	0.8	mA
(V_{in} = 0 to 2.4 V)	SCI		–	0.8	–	1.0	–	1.0	
Input Leakage Current (V_{in} = 0 to 5.25 V)	$\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}	–	2.5	–	5.0	–	5.0	μA
Hi-Z (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	Ports 1, 2, and 3	I_{TSI}	–	10	–	20	–	20	μA
Output High Voltage		V_{OH}							V
(I_{Load} = −65 μA, V_{CC} = Min)*	E, Port 4, SC1, SC2		$V_{SS}+2.4$	–	$V_{SS}+2.4$	–	$V_{SS}+2.4$	–	
(I_{Load} = −100 μA, V_{CC} = Min)	Other Outputs		$V_{SS}+2.4$	–	$V_{SS}+2.4$	–	$V_{SS}+2.4$	–	
Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = Min)	All Outputs	V_{OL}	–	$V_{SS}+0.5$	–	$V_{SS}+0.6$	–	$V_{SS}+0.6$	V
Darlington Drive Current (V_O = 1.5 V)	Port 1	I_{OH}	1.0	4.0	1.0	5.0	1.0	5.0	mA
Internal Power Dissipation (Measured at $T_A = T_L$ in Steady-State Operation)		P_{INT}	–	1200	–	1500	–	1500	mW
Input Capacitance	Port 3, Port 4, SCI	C_{in}	–	12.5	–	12.5	–	12.5	pF
(V_{in} = 0, T_A = 25°C, f_o = 1.0 MHz)	Other Inputs		–	10	–	10	–	10	
V_{CC} Standby	Powerdown	V_{SBB}	4.0	5.25	4.0	5.25	4.0	5.25	V
	Powerup	V_{SB}	4.75	5.25	4.75	5.25	4.75	5.25	
Standby Current	Powerdown	I_{SBB}	–	6.0	–	8.0	–	8.0	mA

*Negotiable to −100 μA (for further information contact the factory)

PERIPHERAL PORT TIMING (Refer to Figures 2-5)

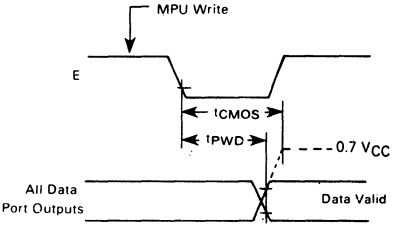
Characteristics	Symbol	EF6801 EF6803		EF6801-1 EF6803-1		EF68A01 EF68A03		EF68B01 EF68B03		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Peripheral Data Setup Time	t_{PDSU}	200	–	200	–	150	–	100	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	200	–	150	–	100	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	t_{PWD}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	2.0	–	2.0	–	2.0	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	200	–	150	–	100	–	ns
Input Data Hold Time	t_{IH}	50	–	50	–	40	–	30	–	ns
Input Data Setup Time	t_{IS}	20	–	20	–	20	–	20	–	ns

FIGURE 2 – DATA SETUP AND HOLD TIMES (MPU READ)



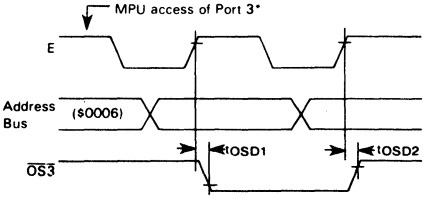
*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

FIGURE 3 – DATA SETUP AND HOLD TIMES (MPU WRITE)



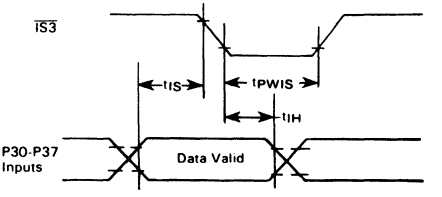
NOTES:
1 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2 Not applicable to P21
3 Port 4 cannot be pulled above V_{CC}

FIGURE 4 – PORT 3 OUTPUT STROBE TIMING (EF6801 SINGLE-CHIP MODE)



*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

FIGURE 5 – PORT 3 LATCH TIMING (EF6801 SINGLE-CHIP MODE)



NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.

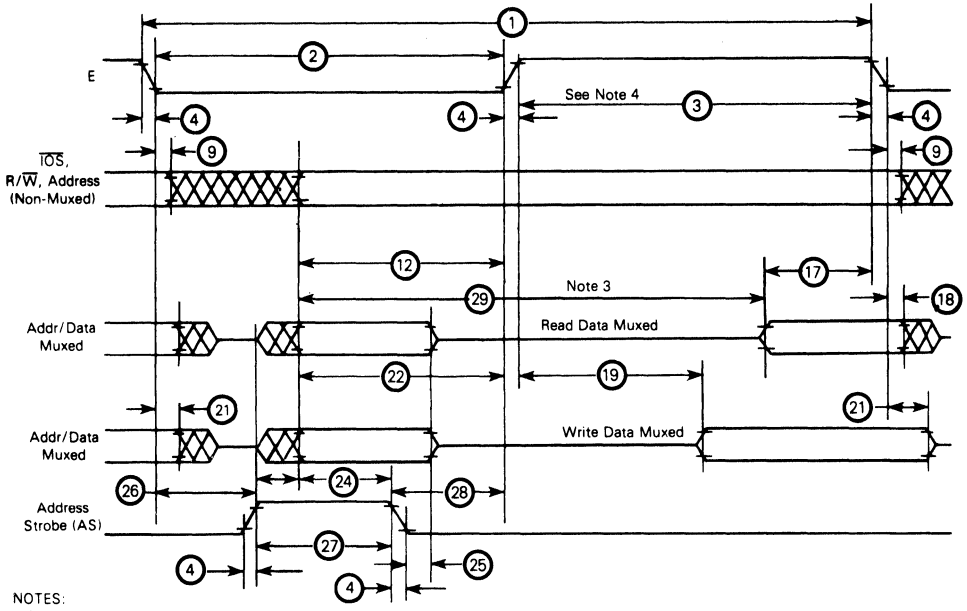
BUS TIMING (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	EF6801 EF6803		EF6801-1 EF6803-1		EF68A01 EF68A03		EF68B01 EF68B03		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	0.667	2.0	0.5	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	300	1000	210	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	300	1000	220	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	20	–	20	–	20	–	10	–	ns
12	Non-Muxed Address Valid Time to E*	t_{AV}	200	–	150	–	115	–	70	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	60	–	40	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	–	170	–	120	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	20	–	10	–	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	200	–	150	–	115	–	80	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	60	–	50	–	40	–	20	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	20	–	20	–	10	–	ns
26	Delay Time, E to AS Rise*	t_{ASD}	90**	–	70**	–	60**	–	45**	–	ns
27	Pulse Width, AS High*	PW_{ASH}	220	–	170	–	140	–	110	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	90	–	70	–	60	–	45	–	ns
29	Usable Access Time*	t_{ACC}	595	–	465	–	380	–	270	–	ns

*At specified cycle time.

**t_{ASD} parameters listed assume external TTL clock drive with 50 % ±5 % duty cycle. Devices driven by an external TTL clock with 50 % ± 1 % duty cycle or which use a crystal have the following t_{ASD} specifications : 100 ns min. (1.0 MHz devices), 80 ns min. (1.25 MHz devices), 65 ns min. (1.5 MHz devices), 50 ns min. (2.0 MHz devices).

FIGURE 6 – BUS TIMING



NOTES:

1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Usable access time is computed by: 12 + 3 – 17 + 4.
4. Memory devices should be enabled only during E high to avoid Port 3 bus contention.

FIGURE 7 – CMOS LOAD

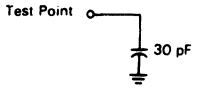
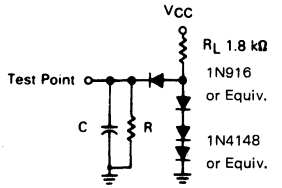


FIGURE 8 – TIMING TEST LOAD PORTS 1, 2, 3, 4



C = 90 pF for P30-P37, P40-P47, E, SC1, SC2
= 30 pF for P10-P17, P20-P24
R = 37 kΩ for P40-P47, E, SC1, SC2
= 24 kΩ for P10-P17, P20-P24
= 24 kΩ for P30-P37

INTRODUCTION

The EF6801 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

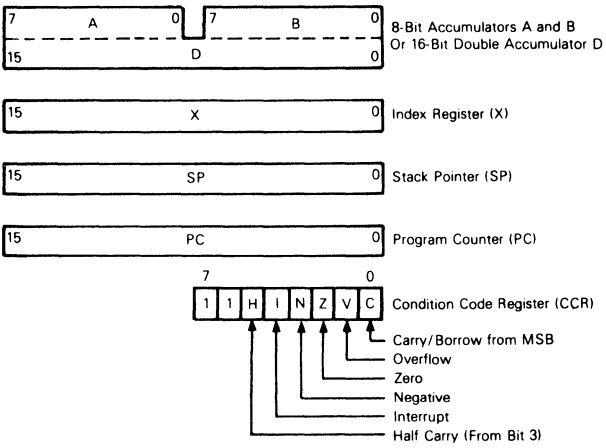
Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a Data Register and a write-only Data Direction Register. The Data Direction Register is used to define whether corresponding bits in the Data Register are configured as an input (clear) or output (set).

The term "port," by itself, refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port," it is controlled by the port Data Direction Register and the programmer has direct access to the port pins using the port Data Register. Port pins are labled as Pij where i identifies one of four ports and j indicates the particular bit.

The Microprocessor Unit (MPU) is an enhanced EF6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the EF6800. The programming model is depicted in Figure 9, where Accumulator D is a concatenation of Accumulators A and B. A list of new operations added to the 6800 instruction set are shown in Table 1.

The EF6803 can be considered an EF6801 that operates in Modes 2 and 3 only.

FIGURE 9 – PROGRAMMING MODEL



OPERATING MODES

The EF6801 provides eight different operating modes (Modes 0 through 7), the EF6803 provides two operating modes (Modes 2 and 3). The operating modes are hardware selectable and determine the device memory map, the configuration of Port 3, Port 4, SC1, SC2, and the physical location of the interrupt vectors.

FUNDAMENTAL MODES

The eight operating modes can be grouped into three fundamental modes which refer to the type of bus it supports: Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Single chip modes include 4 and 7, Expanded Non-Multiplexed is Mode 5 and the remaining five are Expanded Multiplexed modes. Table 2 summarizes the characteristics of the operating modes.

EF6801 Single-Chip Modes (4,7)

In the Single-Chip Mode, the four MCU ports are configured as parallel input/output data ports, as shown in Figure 10. The MCU functions as a monolithic microcomputer in these two modes without external address or data buses. A maximum of 29 I/O lines and two Port 3 control lines are provided. Peripherals or another MCU can be interfaced to Port 3 in a loosely coupled dual processor configuration, as shown in Figure 11.

TABLE 1 — NEW INSTRUCTIONS

Instruction	Description
ABX	Unsigned addition of Accumulator B to Index Register
ADDD	Adds (without carry) the double accumulator to memory and leaves the sum in the double accumulator
ASLD or LSLD	Shifts the double accumulator left (towards MSB) one bit; the LSB is cleared and the MSB is shifted into the C-bit
BHS	Branch if Higher or Same; unsigned conditional branch (same as BCC)
BLO	Branch if Lower; Unsigned conditional branch (same as BCS)
BRN	Branch Never
JSR	Additional addressing mode: direct
LDD	Loads double accumulator from memory
LSL	Shifts memory or accumulator left (towards MSB) one bit; the LSB is cleared and the MSB is shifted into the C-bit (same as ASL)
LSRD	Shifts the double accumulator right (towards LSB) one bit; the MSB is cleared and the LSB is shifted into the C-bit
MUL	Unsigned multiply; multiplies the two accumulators and leaves the product in the double accumulator
PSHX	Pushes the Index Register to stack
PULX	Pulls the Index Register from stack
STD	Stores the double accumulator to memory
SUBD	Subtracts memory from the double accumulator and leaves the difference in the double accumulator
CPX	Internal processing modified to permit its use with any conditional branch instruction

In Single-Chip Test Mode (4), the RAM responds to $XX80 through $XXFF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using modes 0, 1, 2, or 6. If the MCU is Reset and then programmed into Mode 4, execution will begin at $XXFE:XXFF. Mode 5 can be irreversibly entered from Mode 4 without asserting $\overline{\text{RESET}}$ by setting bit 5 of the Port 2 Data Register. This mode is used primarily to test Ports 3 and 4 in the Single-Chip and Non-Multiplexed Modes.

EF6801 Expanded Non-Multiplexed Mode (5)

A modest amount of external memory space is provided in the Expanded Non-Multiplexed Mode while significant on-chip resources are retained. Port 3 functions as an 8-bit bidirectional data bus and Port 4 is configured initially as an input data port. Any combination of the eight least-significant address lines may be obtained by writing to the Port 4 Data Direction Register. Stated alternatively, any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Internal pullup resistors pull the Port 4 lines high until the port is configured.

Figure 12 illustrates a typical system configuration in the Expanded Non-Multiplexed Mode. The MCU interfaces directly with 6800 family parts and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and can be used as a memory page select or chip select line.

TABLE 2 – SUMMARY OF EF6801/03 OPERATING MODES

Common to all Modes:
Reserved Register Area
Port 1
Port 2
Programmable Timer
Serial Communications Interface
Single Chip Mode 7
128 bytes of RAM; 2048 bytes of ROM
Port 3 is a parallel I/O port with two control lines
Port 4 is a parallel I/O port
SC1 is Input Strobe 3 ($\overline{\text{IS3}}$)
SC2 is Output Strobe 3 ($\overline{\text{OS3}}$)
Expanded Non-Multiplexed Mode 5
128 bytes of RAM; 2048 bytes of ROM
256 bytes of external memory space
Port 3 is an 8-bit data bus
Port 4 is an input port/address bus
SC1 is Input/Output Select ($\overline{\text{IOS}}$)
SC2 is Read/Write (R/$\overline{\text{W}}$)
Expanded Multiplexed Modes 1, 2, 3, 6*
Four memory space options (64K address space):
(1) No internal RAM or ROM (Mode 3)
(2) Internal RAM, no ROM (Mode 2)
(3) Internal RAM and ROM (Mode 1)
(4) Internal RAM, ROM with partial address bus (Mode 6)
Port 3 is a multiplexed address/data bus
Port 4 is an address bus (inputs/address in Mode 6)
SC1 is Address Strobe (AS)
SC2 is Read/Write (R/$\overline{\text{W}}$)
Test Modes 0 and 4
Expanded Multiplexed Test Mode 0
May be used to test RAM and ROM
Single Chip and Non-Multiplexed Test Mode 4
(1) May be changed to Mode 5 without going through Reset
(2) May be used to test Ports 3 and 4 as I/O ports

*The EF6803 operates only in modes 2 and 3

FIGURE 10 – SINGLE-CHIP MODE

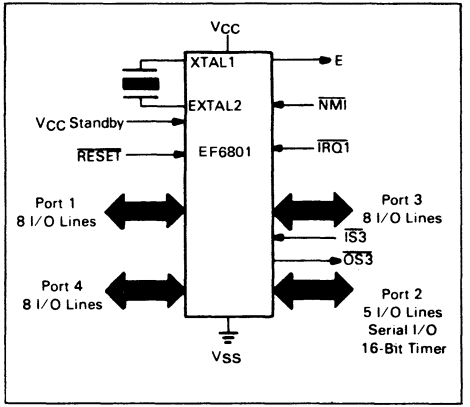


FIGURE 11 – SINGLE-CHIP DUAL PROCESSOR CONFIGURATION

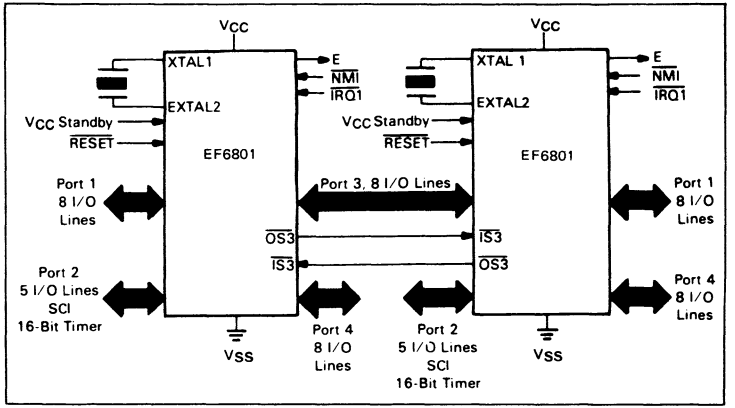
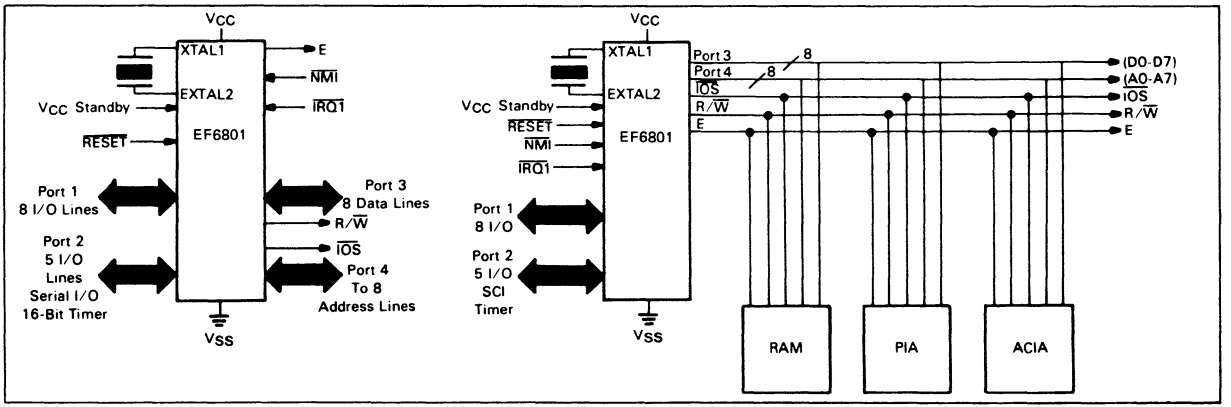


FIGURE 12 – EXPANDED NON-MULTIPLEXED CONFIGURATION



Expanded-Multiplexed Modes (0, 1, 2, 3, 6)

A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes Port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS), and data valid while E is high. In Modes 0 to 3, Port 4 provides address lines A8 to A15. In Mode 6, however, Port 4 initially is configured at $\overline{\text{RESET}}$ as an input data port. The port 4 Data Direction Register can then be changed to provide any combination of address lines, A8 to A15. Stated alternatively, any subset of A8 to A15 can be provided while retaining the remaining port 4 lines as input data lines. Internal pullup resistors pull the Port 4 lines high until software configures the port.

In Mode 0, the Reset vector is external for the first two E-cycles after the positive edge of $\overline{\text{RESET}}$, and internal thereafter. In addition, the internal and external data buses are connected so there must be no memory map overlap in order to avoid potential bus conflicts. Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with the automated test equipment.

Only the EF6801 can operate in each of the expanded multiplexed modes. The EF6803 operates only in Modes 2 and 3.

Figure 13 depicts a typical configuration for the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in Figure 14. This allows Port 3 to function as a Data Bus when E is high.

PROGRAMMING THE MODE

The operating mode is determined at $\overline{\text{RESET}}$ by the levels asserted on P22, P21, and P20. These levels are latched into PC2, PC1, and PC0 of the program control register on the positive edge of $\overline{\text{RESET}}$. The operating mode may be read from the Port 2 Data Register as shown below, and programming levels and timing must be met as shown in Figure 15. A brief outline of the operating modes is shown in Table 3.

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$0003

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in Figure 16 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

TABLE 3 — MODE SELECTION SUMMARY

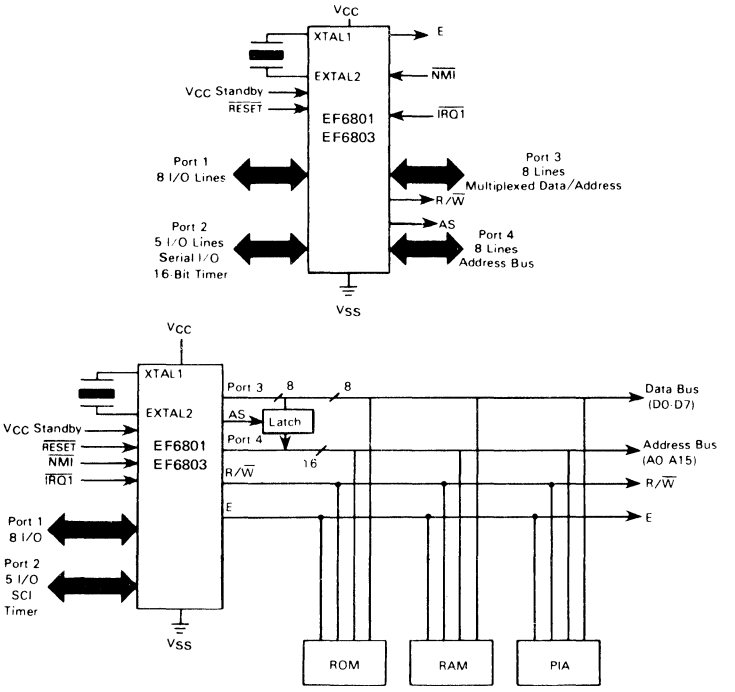
Mode*	P22 PC2	P21 PC1	P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(5, 6)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(5, 6)	Non-Multiplexed/Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX(4)	Multiplexed/No RAM or ROM
2	L	H	L	E	I	E	MUX(4)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM & ROM
0	L	L	L	I	I	I(3)	MUX(4)	Multiplexed Test

Legend:
I — Internal
E — External
MUX — Multiplexed
NMUX — Non-Multiplexed
L — Logic "0"
H — Logic "1"

Notes:
(1) Internal RAM is addressed at $XX80
(2) Internal ROM is disabled
(3) $\overline{\text{RESET}}$ vector is external for 2 cycles after $\overline{\text{RESET}}$ goes high
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
(5) Addresses associated with Port 3 are considered external in Modes 5 and 6
(6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register

*The EF6803 operates only in Modes 2 and 3

FIGURE 13 – EXPANDED MULTIPLEXED CONFIGURATION



NOTE: To avoid data bus (Port 3) contention in the expanded multiplexed modes, memory devices should be enabled only during E high time.

FIGURE 14 – TYPICAL LATCH ARRANGEMENT

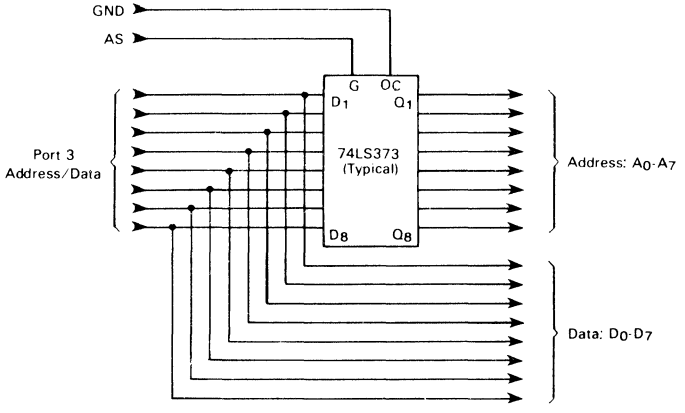
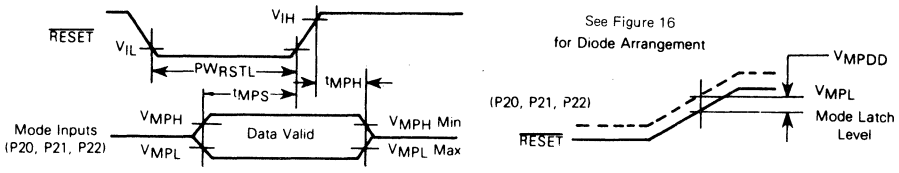


FIGURE 15 – MODE PROGRAMMING TIMING

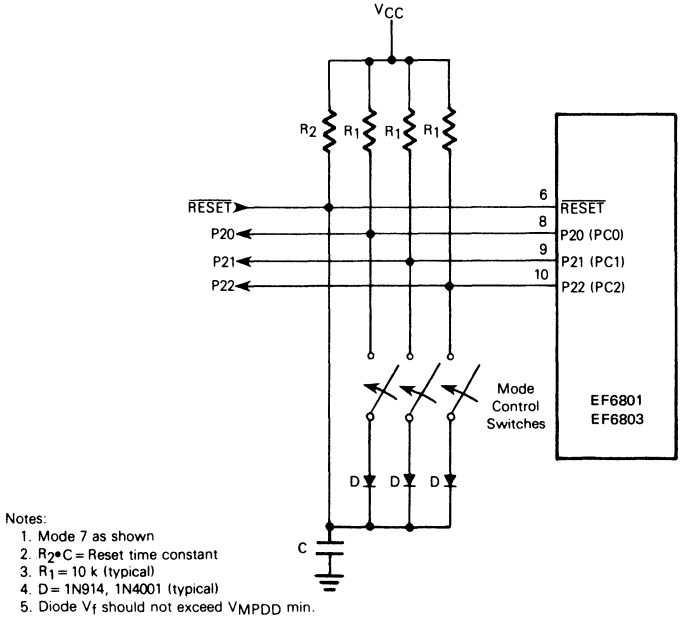


MODE PROGRAMMING (Refer to Figure 15)

Characteristic	Symbol	Min	Max	Unit
Mode Programming Input Voltage Low*	V_{MPL}	–	1.8	V
Mode Programming Input Voltage High	V_{MPH}	4.0	–	V
Mode Programming Diode Differential (If Diodes are Used)	V_{MPDD}	0.6	–	V
$\overline{RESET}$ Low Pulse Width	PW_{RSTL}	3.0	–	E-Cycles
Mode Programming Setup Time	t_{MPS}	2.0	–	E-Cycles
Mode Programming Hold Time $\overline{RESET}$ Rise Time ≥ 1 μs $\overline{RESET}$ Rise Time < 1 μs	t_{MPH}	0 100	– –	ns

*For T_A = –40°C to + 105°C, V_{MPL} = 1.7 V.

FIGURE 16 – TYPICAL MODE PROGRAMMING CIRCUIT



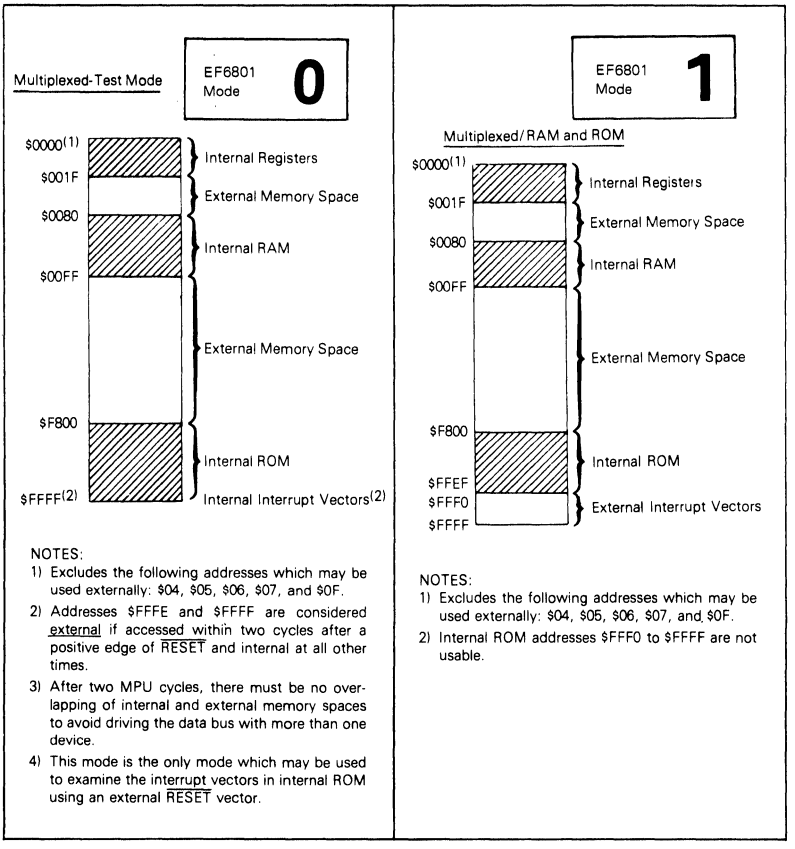
Notes:
1. Mode 7 as shown
2. R_2*C = Reset time constant
3. R_1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min.

MEMORY MAPS

The 6801 Family can provide up to 64K byte address space depending on the operating mode. A memory map for each operating mode is shown in Figure 17.

The first 32 locations of each map are reserved for the internal register area, as shown in Table 4, with exceptions as indicated.

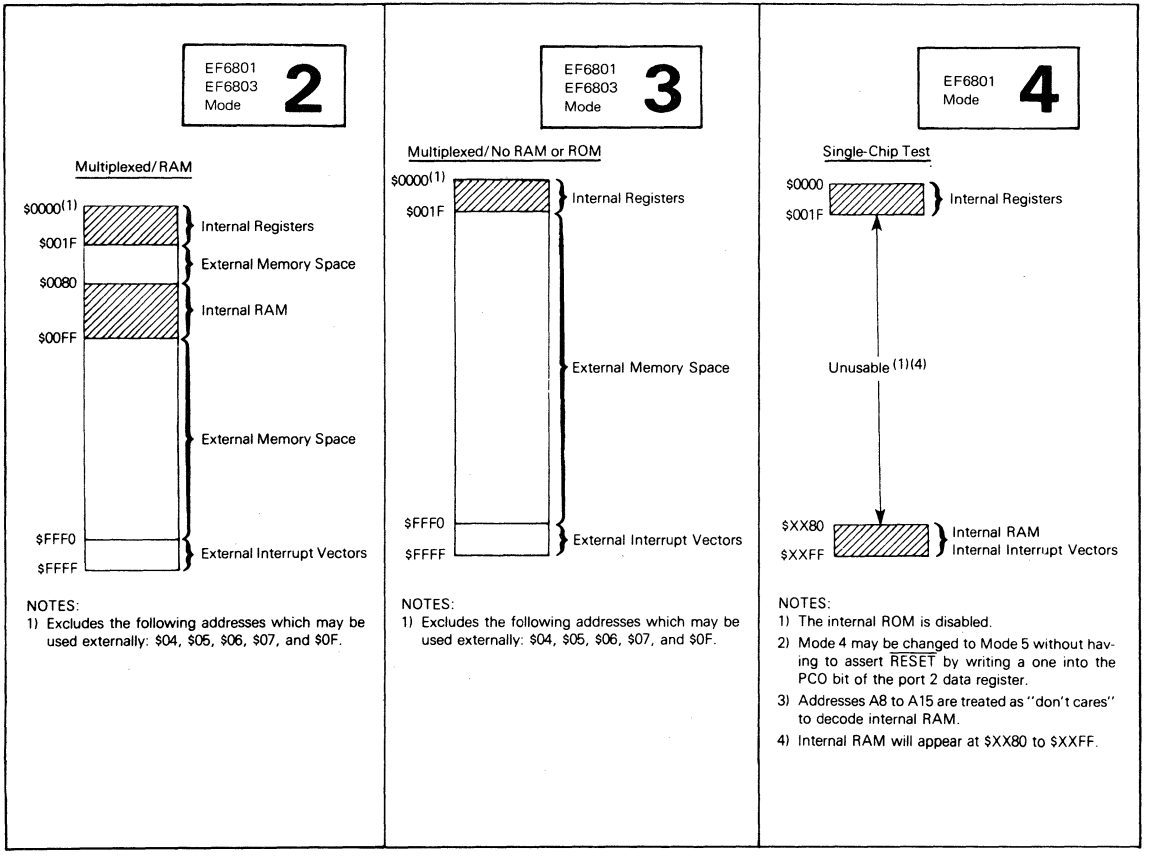
FIGURE 17 – EF6801/03 MEMORY MAPS (Sheet 1 of 3)



NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) Addresses $FFFE and $FFFF are considered external if accessed within two cycles after a positive edge of $\overline{\text{RESET}}$ and internal at all other times.
3) After two MPU cycles, there must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which may be used to examine the interrupt vectors in internal ROM using an external $\overline{\text{RESET}}$ vector.

NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) Internal ROM addresses $FFF0 to $FFFF are not usable.

FIGURE 17 – EF6801/03 MEMORY MAPS (Sheet 2 of 3)



NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

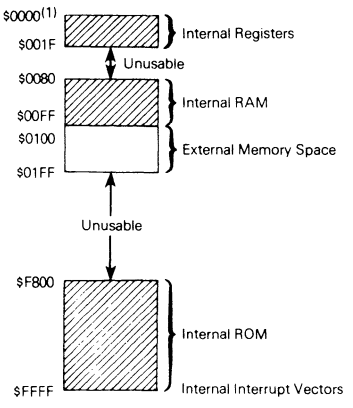
NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

NOTES:
1) The internal ROM is disabled.
2) Mode 4 may be changed to Mode 5 without having to assert $\overline{\text{RESET}}$ by writing a one into the PCO bit of the port 2 data register.
3) Addresses A8 to A15 are treated as "don't cares" to decode internal RAM.
4) Internal RAM will appear at $XX80 to $XXFF.

FIGURE 17 – EF6801/03 MEMORY MAPS (Sheet 3 of 3)

EF6801 Mode **5**

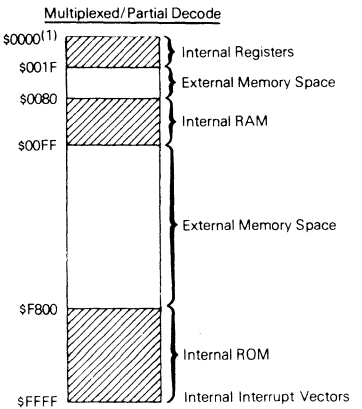
Non-Multiplexed/Partial Decode



NOTES:
1) Excludes the following addresses which may **not** be used externally: $04, $06, and $0F (no $\overline{IOS}$).
2) This mode may be entered without going through $\overline{RESET}$ by using mode 4 and subsequently writing a one into the PCO bit of the port 2 data register.
3) Address lines A0 to A7 will not contain addresses until the data direction register for port 4 has been written with ones in the appropriate bits. These address lines will assert ones until made outputs by writing the data direction register.

EF6801 Mode **6**

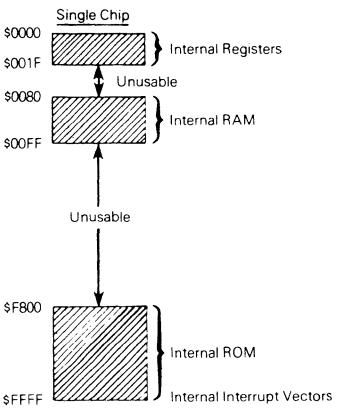
Multiplexed/Partial Decode



NOTES:
1) Excludes the following addresses which may be used externally: $04, $06, and $0F.
2) Address lines A8-A15 will not contain addresses until the data direction register for port 4 has been written with ones in the appropriate bits. These address lines will assert ones until made outputs by writing the data direction register.

EF6801 Mode **7**

Single Chip



EF6801/03 INTERRUPTS

The 6801 Family supports two types of interrupt requests: maskable and non-maskable. A Non-Maskable Interrupt ($\overline{NMI}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the Condition Code Register I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types: $\overline{IRQ1}$ and $\overline{IRQ2}$. The Programmable Timer and Serial Communications Interface use an internal $\overline{IRQ2}$ interrupt line, as shown in Figure 1. External devices (and IS3) use $\overline{IRQ1}$. An $\overline{IRQ1}$ interrupt is serviced before $\overline{IRQ2}$ if both are pending.

All IRQ2 interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order and each is vectored to a separate location. All interrupt vector locations are shown in Table 5.

The Interrupt flowchart is depicted in Figure 18 and is common to every interrupt excluding reset. During interrupt servicing the Program Counter, Index Register, A Accumulator, B Accumulator, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. Interrupt and $\overline{RESET}$ timing are illustrated in Figures 19 and 20.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power to a large portion of the MCU. The power supply should provide +5 volts (±5%) to V_{CC}, and V_{SS} should be tied to ground. Total power dissipation (including V_{CC} Standby), will not exceed P_D milliwatts.

V_{CC} STANDBY

V_{CC} Standby provides power to the standby portion ($80 through $BF) of the RAM and the STBY PWR and RAME bits of the RAM Control Register. Voltage requirements depend on whether the device is in a powerup or powerdown state. In the powerup state, the power supply should provide +5 volts (±5%) and must reach V_{SB} volts before $\overline{RESET}$ reaches 4.0 volts. During powerdown, V_{CC} Standby must remain above V_{SBB} (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} Standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during powerdown operation. V_{CC} Standby should be tied to ground in Mode 3.

TABLE 4 — INTERNAL REGISTER AREA

Register	Address
Port 1 Data Direction Register***	00
Port 2 Data Direction Register***	01
Port 1 Data Register	02
Port 2 Data Register	03
Port 3 Data Direction Register***	04*
Port 4 Data Direction Register***	05**
Port 3 Data Register	06*
Port 4 Data Register	07**
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 Control and Status Register	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

*External addresses in Modes 0, 1, 2, 3, 5, 6; cannot be accessed in Mode 5 (No $\overline{IOS}$)
**External addresses in Modes 0, 1, 2, 3
***1 = Output, 0 = Input

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

MSB	LSB	Interrupt
FFFE	FFFF	$\overline{RESET}$
FFFC	FFFD	$\overline{NMI}$
FFFA	FFFB	Software Interrupt (SWI)
FFF8	FFF9	$\overline{IRQ1}$ (or $\overline{IS3}$)
FFF6	FFF7	ICF (Input Capture)*
FFF4	FFF5	OCF (Output Compare)*
FFF2	FFF3	TOF (Timer Overflow)*
FFF0	FFF1	SCI (RDRF + ORFE + TDRE)*

*IRQ2 Interrupt

FIGURE 18 – INTERRUPT FLOWCHART

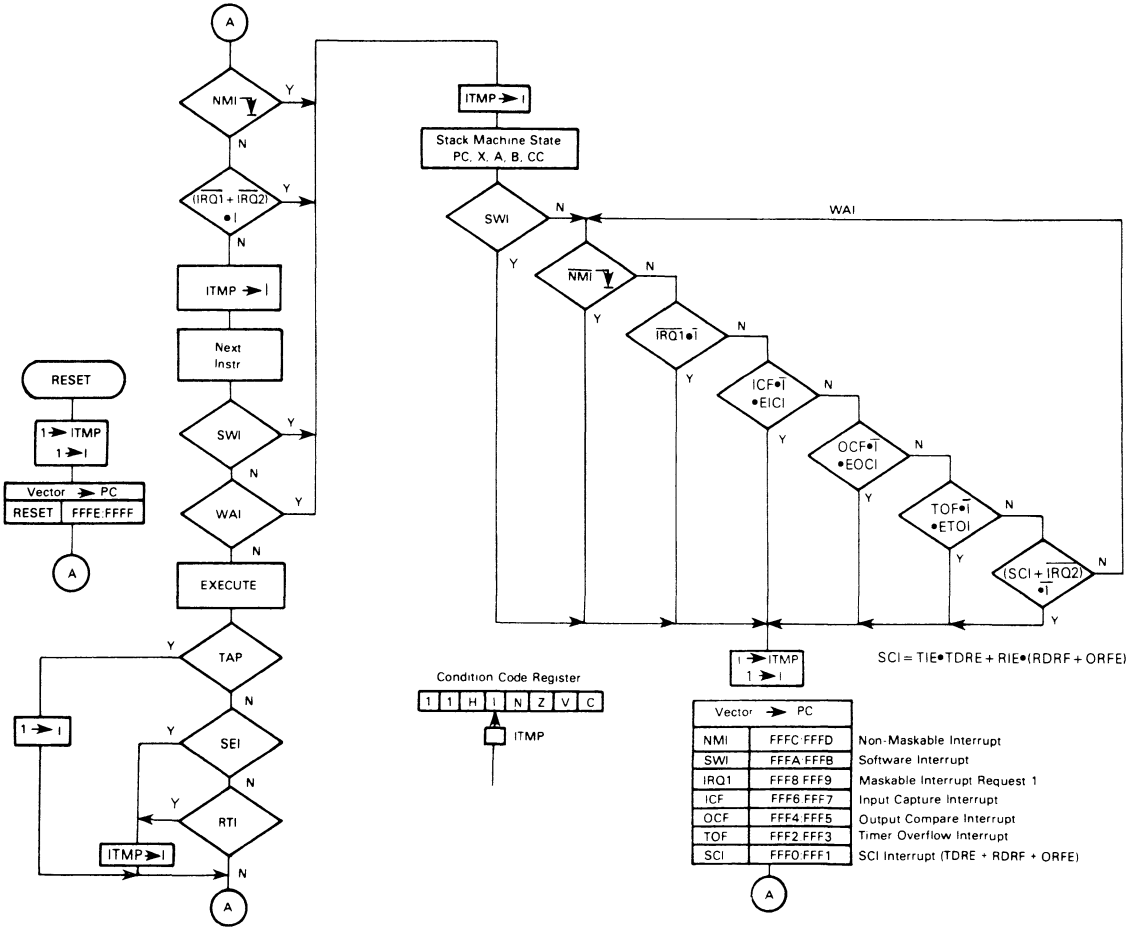


FIGURE 19 – INTERRUPT SEQUENCE

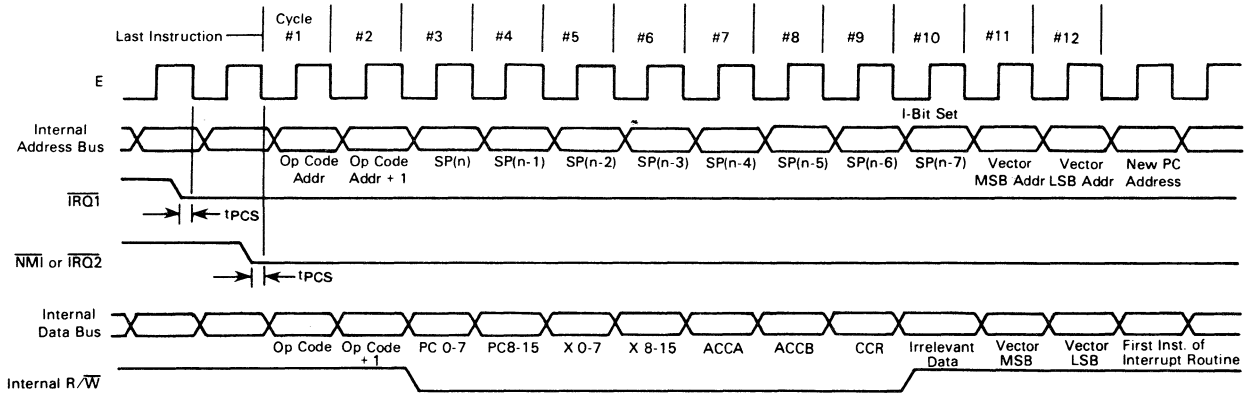
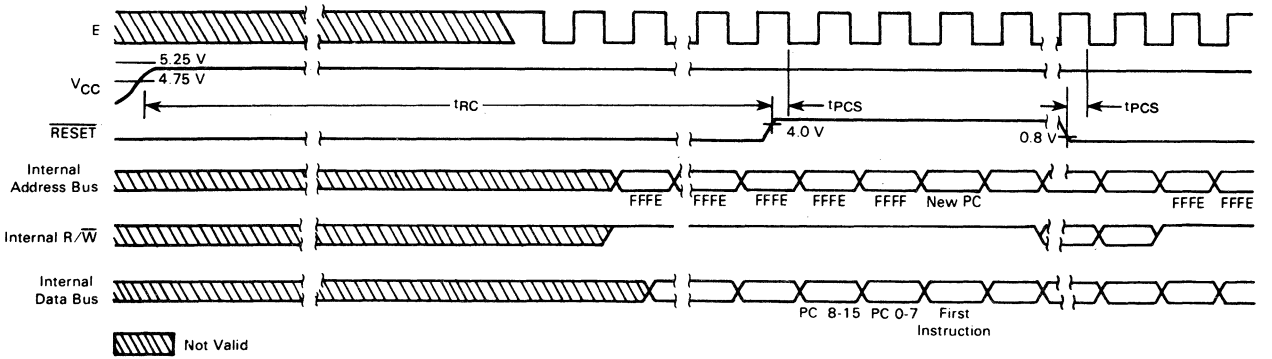


FIGURE 20 – RESET TIMING



XTAL1 AND EXTAL2

These two input pins interface either a crystal or TTL compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58 MHz or 4.4336 MHz Color Burst TV crystals. A 20 pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL compatible clock at $4f_O$ with a duty cycle of 50% (±5%) with XTAL1 connected to ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for f_{XTAL}. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.* The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in Figure 21.

$\overline{\text{RESET}}$

This input is used to reset the internal state of the device and provide an orderly startup procedure. During powerup, $\overline{\text{RESET}}$ must be held below 0.8 volts: (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} Standby reaches 4.75 volts. $\overline{\text{RESET}}$ must be held low at least three E-cycles if asserted during powerup operation.

E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the device input clock frequency. It will drive one Schottky TTL load and 90 pF, and all data given in cycles is referenced to this clock unless otherwise noted.

$\overline{\text{NMI}}$ (NON-MASKABLE INTERRUPT)

An $\overline{\text{NMI}}$ negative edge requests an MCU interrupt sequence, but the current instruction will be completed before it responds to the request. The MCU will then begin an interrupt sequence. Finally, a vector is fetched from $FFFC and $FFFD, transferred to the Program Counter and instruction execution is resumed. $\overline{\text{NMI}}$ typically requires a 3.3 kΩ (nominal) resistor to V_{CC}. There is no internal $\overline{\text{NMI}}$ pullup resistor. $\overline{\text{NMI}}$ must be held low for at least one E-cycle to be recognized under all conditions.

$\overline{\text{IRQ1}}$ (MASKABLE INTERRUPT REQUEST 1)

$\overline{\text{IRQ1}}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the MCU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9, transferred to the Program Counter, and instruction execution is resumed.

$\overline{\text{IRQ1}}$ typically requires an external 3.3 kΩ (nominal) resistor to V_{CC} for wire-OR applications. $\overline{\text{IRQ1}}$ has no internal pullup resistor.

SC1 AND SC2 (STROBE CONTROL 1 AND 2)

The function of SC1 and SC2 depends on the operating mode. SC1 is configured as an output in all modes except single chip mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90 pF.

SC1 and SC2 In Single-Chip Mode

In Single-Chip Mode, SC1 and SC2 are configured as an input and output, respectively, and both function as Port 3 control lines. SC1 functions as $\overline{\text{IS3}}$ and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with $\overline{\text{IS3}}$ are controlled by Port 3 Control and Status Register and are discussed in the Port 3 description. If unused, $\overline{\text{IS3}}$ can remain unconnected.

SC2 is configured as $\overline{\text{OS3}}$ and can be used to strobe output data or acknowledge input data. It is controlled by Output Strobe Select (OSS) in the Port 3 Control and Status Register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the Port 3 Data Register. $\overline{\text{OS3}}$ timing is shown in Figure 4.

SC1 And SC2 In Expanded Non-Multiplexed Mode

In the Expanded Non-Multiplexed Mode, both SC1 and SC2 are configured as outputs. SC1 functions as Input/Output Select ($\overline{\text{IOS}}$) and is asserted only when $0100 through $01FF is sensed on the internal address bus.

SC2 is configured as Read/Write and is used to control the direction of data bus transfers. An MPU read is enabled when Read/Write and E are high.

SC1 And SC2 In Expanded Multiplexed Mode

In the Expanded Multiplexed Modes, both SC1 and SC2 are configured as outputs. SC1 functions as Address Strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by Address Strobe captures address on the negative edge, as shown in Figure 14.

SC2 is configured as Read/Write and is used to control the direction of data bus transfers. An MPU read is enabled when Read/Write and E are high.

P10-P17 (PORT 1)

Port 1 is a mode independent 8-bit I/O port with each line an input or output as defined by the Port 1 Data Direction Register. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 pF, Darlington transistors, or CMOS devices using external pullup resistors. It is configured as a data input port by $\overline{\text{RESET}}$. Unused lines can remain unconnected.

P20-P24 (PORT 2)

Port 2 is a mode-independent, 5-bit, multipurpose I/O port. The voltage levels present on P20, P21, and P22 on the rising edge of $\overline{\text{RESET}}$ determine the operating mode of the MCU. The entire port is then configured as a data input port. The Port 2 lines can be selectively configured as data output lines by setting the appropriate bits in the Port 2 Data Direction Register. The Port 2 Data Register is used to move data through the port. However, if P21 is configured as an output, it will be tied to the timer Output Compare function and cannot be used to provide output from the Port 2 Data Register.

Port 2 can also be used to provide an interface for the Serial Communications Interface and the timer Input Edge function. These configurations are described in the Programmable Timer and Serial Communications Interface (SCI) section.

The Port 2 three-state, TTL-compatible output buffers are capable of driving one Schottky TTL load and 30 pF, or CMOS devices using external pullup resistors.

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$0003

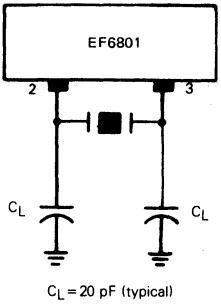
FIGURE 21 – 6801 FAMILY OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

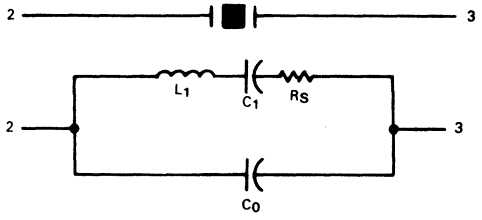
Nominal Crystal Parameters*

	3.58 MHz	4.00 MHz	5.0 MHz	6.0 MHz	8.0 MHz
RS	60 Ω	50 Ω	30-50 Ω	30-50 Ω	20-40 Ω
C_0	3.5 pF	6.5 pF	4-6 pF	4-6 pF	4-6 pF
C_1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF	0.01-0.02 pF
Q	> 40 K	> 30 K	> 20 K	> 20 K	> 20 K

*NOTE: These are representative AT-cut crystal parameters only. Crystals of other types of cut may also be used.

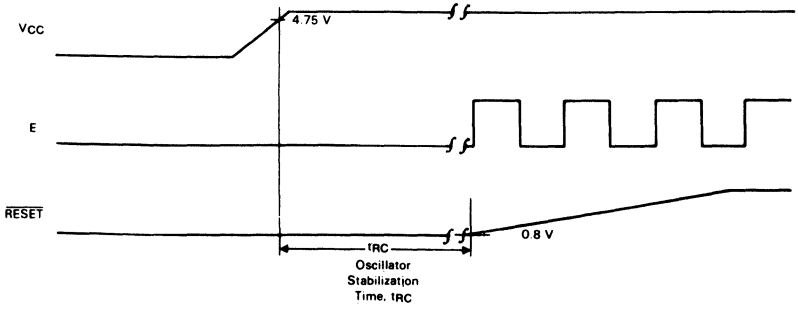


C_L = 20 pF (typical)



Equivalent Circuit

(b) Oscillator Stabilization Time (t_{RC})



P30-P37 (PORT 3)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address/data bus depending on the operating mode. The TTL compatible three-state output buffers can drive one Schottky TTL load and 90 pF. Unused lines can remain unconnected.

Port 3 In Single-Chip Mode

Port 3 is an 8-bit I/O port in the Single-Chip Mode, with each line configured by the Port 3 Data Direction Register. There are also two lines, $\overline{IS3}$ and $\overline{OS3}$, which can be used to control Port 3 data transfers.

Three Port 3 options are controlled by the Port 3 Control and Status Register and are available only in Single-Chip Mode: (1) Port 3 input data can be latched using $\overline{IS3}$ as a control signal, (2) $\overline{OS3}$ can be generated by either an MPU read or write to the Port 3 Data Register, and (3) an $\overline{IRQ1}$ interrupt can be enabled by an $\overline{IS3}$ negative edge. Port 3 latch timing is shown in Figure 5.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3 Flag	IS3 IRQ1 Enable	X	OSS	Latch Enable	X	X	X	$000F

Bit 0-2 Not used.

Bit 3 LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an $\overline{IS3}$ negative edge. The latch is transparent after a read of the Port 3 Data Register. LATCH ENABLE is cleared during reset.

Bit 4 OSS (Output Strobe Select). This bit determines whether $\overline{OS3}$ will be generated by a read or write of the Port 3 Data Register. When clear, the strobe is generated by a read; when set, it is generated by a write. OSS is cleared during reset.

Bit 5 Not used.

Bit 6 IS3 IRQ1 ENABLE. When set, an $\overline{IRQ1}$ interrupt will be enabled whenever IS3 FLAG is set; when clear, the interrupt is inhibited. This bit is cleared during reset.

Bit 7 IS3 FLAG. This read-only status bit is set by an $\overline{IS3}$ negative edge. It is cleared by a read of the Port 3 Control and Status Register (with IS3 FLAG set) followed by a read or write to the Port 3 Data Register or during reset.

Port 3 In Expanded Non-Multiplexed Mode

Port 3 is configured as a bidirectional data bus (D7-D0) in the Expanded Non-Multiplexed Mode. The direction of data transfers is controlled by Read/Write (SC2). Data is clocked by E (Enable).

Port 3 In Expanded Multiplexed Mode

Port 3 is configured as a time multiplexed address (A0-A7) and data bus (D7-D0) in the Expanded Multiplexed Modes, where Address Strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high impedance state between valid address and data to prevent bus conflicts.

P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF and is the only port with internal pullup resistors. Unused lines can remain unconnected.

Port 4 In Single-Chip Mode

In Single-Chip Mode, Port 4 functions as an 8-bit I/O port with each line configured by the Port 4 Data Direction Register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

Port 4 In Expanded Non-Multiplexed Mode

Port 4 is configured from reset as an 8-bit input port, where the Port 4 Data Direction Register can be written to provide any or all of eight address lines, A0 to A7. Internal pullup resistors pull the lines high until the Port 4 Data Direction Register is configured.

Port 4 In Expanded Multiplexed Mode

In all Expanded Multiplexed modes except Mode 6, Port 4 functions as half of the address bus and provides A8 to A15. In Mode 6, the port is configured from reset as an 8-bit parallel input port, where the Port 4 Data Direction Register can be written to provide any or all of upper address lines A8 to A15. Internal pullup resistors pull the lines high until the Port 4 Data Direction Register is configured, where bit 0 controls A8.

RESIDENT MEMORY

The EF6801 provides 2048 bytes of on-board ROM and 128 bytes of on-board RAM.

One half of the RAM is powered through the V_{CC} standby pin and is maintainable during V_{CC} powerdown. This standby portion of the RAM consists of 64 bytes located from $80 through $BF.

Power must be supplied to V_{CC} standby if the internal RAM is to be used regardless of whether standby power operation is anticipated.

The RAM is controlled by the RAM Control Register.

RAM CONTROL REGISTER ($14)

The RAM Control Register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during powerdown operation. It is intended that RAME be cleared and STBY PWR be set as part of a powerdown procedure.

RAM CONTROL REGISTER

7	6	5	4	3	2	1	0
STBY PWR	RAME	X	X	X	X	X	X

Bit 0-5 Not used.

Bit 6 RAME RAM Enable. This read/write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of $\overline{\text{RESET}}$. If RAME is clear, any access to a RAM address is external. If RAME is set and not in mode 3, the RAM is included in the internal map.

Bit 7 STBY PWR Standby Power. This bit is a read/write status bit which, when once set, remains set as long as V_{CC} standby remains above V_{SBB} (minimum). As long as this bit is set following a period of standby operation, the standby power supply has adequately preserved the data in the standby RAM. If this bit is cleared during a period of standby operation, it indicates that V_{CC} standby had fallen to a level sufficiently below V_{SBB} (minimum) to suspect that data in the standby RAM is not valid. This bit can be set only by software and is not affected during reset.

PROGRAMMABLE TIMER

The programmable timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 22.

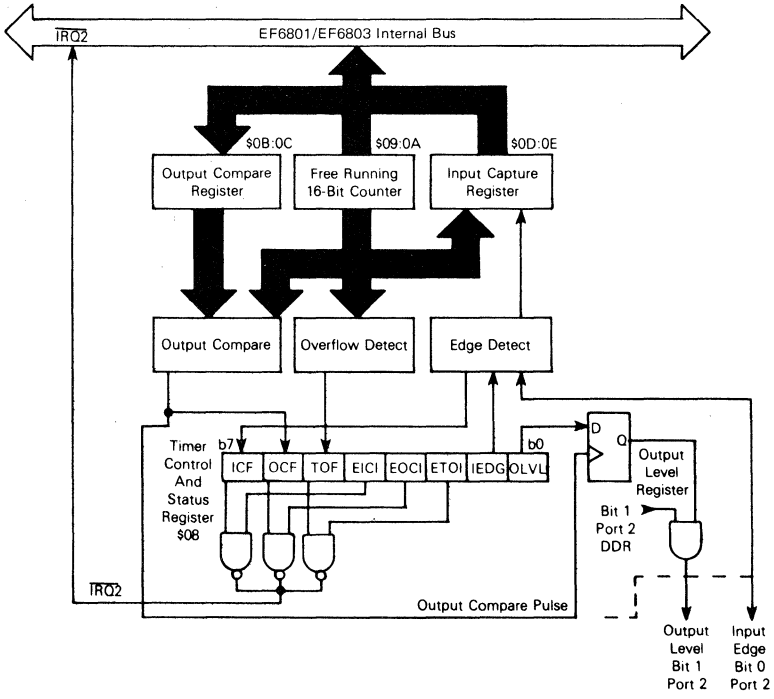
COUNTER ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (enable). It is cleared during reset and is read-only with one exception: a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all ones.

OUTPUT COMPARE REGISTER ($0B:0C)

The output compare register is a 16-bit read/write register used to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E cycle. When a match occurs, OCF is set and OLVL is clocked to an output level register. If port 2, bit 1, is configured as an output, OLVL will appear at P21 and the output compare register and OLVL can then be changed for the next

FIGURE 22 – BLOCK DIAGRAM OF PROGRAMMABLE TIMER



and OLVL is clocked to an output level register. If Port 2, bit 1, is configured as an output, OLVL will appear at P21 and the Output Compare Register and OLVL can then be changed for the next compare. The function is inhibited for one cycle after a write to its high byte ($0B) to ensure a valid compare. The Output Compare Register is set to $FFFF at $\overline{\text{RESET}}$.

INPUT CAPTURE REGISTER ($0D:0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20 even when configured as an output. An input capture can occur independently of ICF: the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTER ($08)

The Timer Control and Status Register (TCSR) is an 8-bit register of which all bits are readable, while only bits 0-4 can be written. The three most significant bits provide the timer status and indicate if:

- a proper level transition has been detected,
- a match has occured between the free-running counter and the output compare register, and
- the free-running counter has overflowed.

Each of the three events can generate an $\overline{\text{IRQ2}}$ interrupt and is controlled by an individual enable bit in the TCSR.

TIMER CONTROL AND STATUS REGISTER (TCSR)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

Bit 0 OLVL — Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P21 if Bit 1 of the Port 2 Data Direction Register is set. It is cleared during reset.

Bit 1 EIDG — Input Edge. IEDG is cleared during reset and controls which level transition will trigger a counter transfer to the Input Capture Register:

IEDG = 0 Transfer on a negative-edge

IEDG = 1 Transfer on a positive-edge.

Bit 2 ETOI — Enable Timer Overflow Interrupt. When set, an IRQ2 interrupt is enabled for a timer overflow; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 3 EOCI — Enable Output Compare Interrupt. When set, an IRQ2 interrupt is enabled for an output compare; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 4 EICI — Enable Input Capture Interrupt. When set, an IRQ2 interrupt is enabled for an input capture; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 5 TOF — Timer Overflow Flag. TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) then reading the counter high byte ($09), or during reset.

Bit 6 OCF — Output Compare Flag. OCF is set when the Output Compare Register matches the free-running counter. It is cleared by reading the TCSR (with OCF set) and then writing to the Output Compare Register ($0B or $0C), or during reset.

Bit 7 ICF — Input Capture Flag. ICF is set to indicate a proper level transition; it is cleared by reading the TCSR (with ICF set) and then the Input Capture Register High Byte ($0D), or during reset.

SERIAL COMMUNICATIONS INTERFACE (SCI)

A full-duplex asynchronous Serial Communications Interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and Bi-phase and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

In a typical serial loop multi-processor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPU's to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 1's or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable:

- format: standard mark/space (NRZ) or Bi-phase
- clock: external or internal bit rate clock
- Baud: one of 4 per E-clock frequency, or external clock (×8 desired baud)
- wake-up feature: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The Serial Communications Interface includes four addressable registers as depicted in Figure 23. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and a read-only Receive Register. The shift registers are not accessible to software.

Rate and Mode Control Register (RMCR) ($10)

The Rate and Mode Control Register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of four write-only bits which are cleared during reset. The two least significant bits control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER (RMCR)

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	$0010

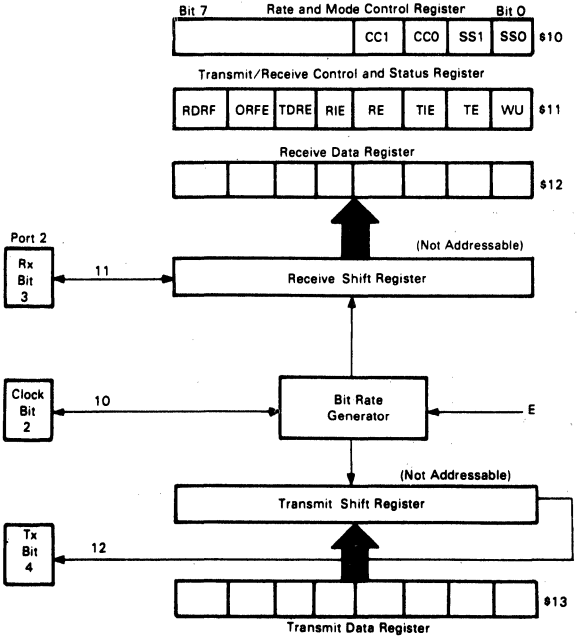
Bit 1:Bit 0 — SS1:SS0 Speed Select. These two bits select the Baud rate when using the internal clock. Four rates may be selected which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

Bit 3:Bit 2 — CC1:CC0 Clock Control and Format Select. These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P22 at eight times (8X) the desired bit rate, but not greater than E, with a duty cycle of 50% (±10%). If CC1:CC0=10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE: The source of SCI internal bit rate clock is the timer free running counter. An MPU write to the counter can disturb serial operations.

FIGURE 23 – SCI REGISTERS



Transmit/Receive Control And Status Register (TRCSR) ($11)

The Transmit/Receive Control and Status Register controls the transmitter, receiver, wake-up feature, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by RESET.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$0011

Bit 0 WU — "Wake-up" on Idle Line. When set, WU enables the wake-up function; it is cleared by ten consecutive 1's or during reset. WU will not set if the line is idle.

Bit 1 TE — Transmit Enable. When set, P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive 1's is transmitted. TE is cleared during reset.

Bit 2 TIE — Transmit Interrupt Enable. When set, an IRQ2 interrupt is enabled when TDRE is set; when clear, the interrupt is inhibited. TE is cleared during reset.

Bit 3 RE — Receive Enable. When set, the P23 DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared during reset.

Bit 4 RIE — Receiver Interrupt Enable. When set, an IRQ2 interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared during reset.

Bit 5 TDRE — Transmit Data Register Empty. TDRE is set when the Transmit Data Register is transferred to the output serial shift register or during reset. It is cleared by reading the TRCSR (with TDRE set) and then writing to the Transmit Data Register. Additional data will be transmitted only if TDRE has been cleared.

Bit 6 ORFE — Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the Receiver Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF: if RDRF is set, then an overrun has occurred; otherwise a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun condition. Unframed data causing a framing error is transferred to the Receive Data Register. However, subsequent data transfer is blocked until the framing error flag is cleared.* ORFE is cleared by reading the TRCSR (with ORFE set) then the Receive Data Register, or during reset.

Bit 7 RDRF — Receive Data Register Full. RDRF is set when the input serial shift register is transferred to the Receive Data Register. It is cleared by reading the TRCSR (with RDRF set), and then the Receive Data Register, or during reset.

TABLE 6 — SCI BIT TIMES AND RATES

SS1:SS0		$4f_O \rightarrow$	2.4576 MHz	4.0 Mhz	4.9152 MHz
		E	614.4 kHz	1.0 MHz	1.2288 MHz
0	0	÷16	26 μs/38,400 Baud	16 μs/62,500 Baud	13.0 μs/76,800 Baud
0	1	÷128	208 μs/4,800 baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1	0	÷1024	1.67 ms/600 Baud$	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1	1	÷4096	6.67 ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud
*External (P22)			13.0 μs/76,800 Baud	8.0 μs/125,000 Baud	6.5 μs/153,600 Baud

*Using maximum clock rate

TABLE 7 — SCI FORMAT AND CLOCK SOURCE CONTROL

CC1:CC0	Format	Clock Source	Port 2 Bit 2
00	Bi-Phase	Internal	Not Used
01	NRZ	Internal	Not Used
10	NRZ	Internal	Output
11	NRZ	External	Input

SERIAL OPERATIONS

The SCI is initialized by writing control bytes first to the Rate and Mode Control Register and then to the Transmit/Receive Control and Status Register. When TE is set, the output of the transmit serial shift register is connected to P24 and serial output is initiated by transmitting a 9-bit preamble of 1's.

At this point one of two situations exist: 1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of 1's will be sent indicating an idle line, or 2) if a byte has been written to the Transmit-Data Register (TDRE = 0), it will be transferred to the output serial shift register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0) and a stop bit (1), will be transmitted. If TDRE is still set when the next byte transfer should occur, 1's will be sent until more data is provided. In Bi-phase format, the output toggles at the start of each bit and at half-bit time when a "1" is sent. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. SCI data formats are illustrated in Figure 24.

INSTRUCTION SET

The EF6801/03 is upward source and object code compatible with the EF6800. Execution times of key instructions have been reduced and several new instructions have been added, including a hardware multiply. A list of new operations added to the EF6800 instruction set is shown in Table 1.

In addition, two new special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the Program Counter to increment like a 16-bit counter, causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in Table 8. There are 220 valid machine codes, 34 unassigned codes, and 2 codes reserved for test purposes.

PROGRAMMING MODEL

A programming model for the EF6801/03 is shown in Figure 10. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

Program Counter – The program counter is a 16-bit register which always points to the next instruction.

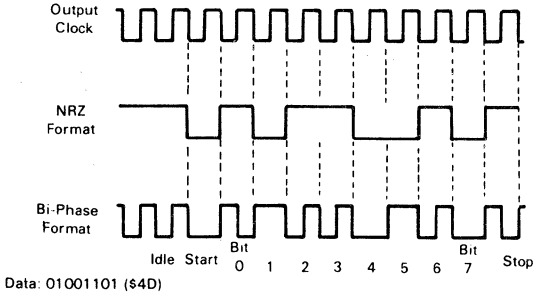
Stack Pointer – The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location defined by the programmer.

Index Register – The Index Register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

Accumulators – The MPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

Condition Code Registers – The condition code register indicates the results of an instruction and includes the following five condition bits: Negative (N), Zero (Z), Overflow (V), Carry/Borrow from MSB (C), and Half Carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7, are read as ones.

FIGURE 24 – SCI DATA FORMATS



ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of addressing modes for all instructions is presented in Tables 9, 10, 11, and 12, where execution times are provided in E-cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 14 and a description of selected instructions is shown in Figure 25.

Immediate Addressing – The operand or "immediate byte(s)" is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

Direct Addressing – The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, the 256-byte area is reserved for frequently referenced data.

Extended Addressing – The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instrutions.

Indexed Addressing – The unsigned offset contained in the second byte of the instruction is added with carry to the Index Register and used to reference memory without changing the Index Register. These are two byte instructions.

Inherent Addressing – The operand(s) are registers and no memory reference is required. These are single byte instructions.

Relative Addressing – Relative addressing is used only for branch instructions. If the branch condition is true, the Program Counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current Program Counter. This provides a branch range of –126 to 129 bytes from the first byte of the instruction. These are two byte instructions.

TABLE 8 – CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#
00	*			
01	NOP	INHER	2	1
02	*			
03	*			
04	LSRD		3	1
05	ASLD		3	1
06	TAP		2	1
07	TPA		2	1
08	INX		3	1
09	DEX		3	1
0A	CLV		2	1
0B	SEV		2	1
0C	CLC		2	1
0D	SEC		2	1
0E	CLI		2	1
0F	SEI		2	1
10	SBA		2	1
11	CBA		2	1
12	*			
13	*			
14	*			
15	*			
16	TAB		2	1
17	TBA		2	1
18	*			
19	DAA	INHER	2	1
1A	*			
1B	ABA	INHER	2	1
1C	*			
1D	*			
1E	*			
1F	*			
20	BRA	REL	3	2
21	BRN		3	2
22	BHI		3	2
23	BLS		3	2
24	BCC		3	2
25	BCS		3	2
26	BNE		3	2
27	BEQ		3	2
28	BVC		3	2
29	BVS		3	2
2A	BPL		3	2
2B	BMI		3	2
2C	BGE		3	2
2D	BLT		3	2
2E	BGT		3	2
2F	BLE	REL	3	2
30	TSX	INHER	3	1
31	INS		3	1
32	PULA		4	1
33	PULB		4	1
34	DES	INHER	3	1
35	TXS		3	1
36	PSHA		3	1
37	PSHB		3	1
38	PULX		5	1
39	RTS		5	1
3A	ABX		3	1
3B	RTI		10	1
3C	PSHX		4	1
3D	MUL		10	1
3E	WAI		9	1
3F	SWI		12	1
40	NEGA		2	1
41	*			
42	*			
43	COMA		2	1
44	LSRA		2	1
45	*			
46	RORA		2	1
47	ASRA		2	1
48	ASLA		2	1
49	ROLA		2	1
4A	DECA		2	1
4B	*			
4C	INCA		2	1
4D	TSTA		2	1
4E	T			
4F	CLRA		2	1
50	NEGB		2	1
51	*			
52	*			
53	COMB		2	1
54	LSRB		2	1
55	*			
56	RORB		2	1
57	ASRB		2	1
58	ASLB		2	1
59	ROLB		2	1
5A	DECB		2	1
5B	*			
5C	INCB		2	1
5D	TSTB		2	1
5E	T			
5F	CLRB	INHER	2	1
60	NEG	INDXD	6	2
61	*			
62	*			
63	COM		6	2
64	LSR		6	2
65	*			
66	ROR		6	2
67	ASR	INDXD	6	2
68	ASL	INDXD	6	2
69	ROL		6	2
6A	DEC		6	2
6B	*			
6C	INC		6	2
6D	TST		6	2
6E	JMP		3	2
6F	CLR	INDXD	6	2
70	NEG	EXTND	6	3
71	*			
72	*			
73	COM		6	3
74	LSR		6	3
75	*			
76	ROR		6	3
77	ASR		6	3
78	ASL		6	3
79	ROL		6	3
7A	DEC		6	3
7B	*			
7C	INC		6	3
7D	TST		6	3
7E	JMP		3	3
7F	CLR	EXTND	6	3
80	SUBA	IMMED	2	2
81	CMPA		2	2
82	SBCA		2	2
83	SUBD		4	3
84	ANDA		2	2
85	BITA		2	2
86	LDAA		2	2
87	*			
88	EORA		2	2
89	ADCA		2	2
8A	ORAA		2	2
8B	ADDA		2	2
8C	CPX	IMMED	4	3
8D	BSR	REL	6	2
8E	LDS	IMMED	3	3
8F	*			
90	SUBA	DIR	3	2
91	CMPA		3	2
92	SBCA		3	2
93	SUBD		5	2
94	ANDA		3	2
95	BITA		3	2
96	LDAA		3	2
97	STAA		3	2
98	EORA		3	2
99	ADCA		3	2
9A	ORAA		3	2
9B	ADDA		3	2
9C	CPX	DIR	5	2
9D	JSR		4	2
9E	LDS		4	2
9F	STS	DIR	4	2
A0	SUBA	INDXD	4	2
A1	CMPA		4	2
A2	SBCA		4	2
A3	SUBD		6	2
A4	ANDA		4	2
A5	BITA		4	2
A6	LDAA		4	2
A7	STAA		4	2
A8	EORA		4	2
A9	ADCA		4	2
AA	ORAA		4	2
AB	ADDA		4	2
AC	CPX		6	2
AD	JSR		6	2
AE	LDS		5	2
AF	STS	INDXD	5	2
B0	SUBA	EXTND	4	3
B1	CMPA		4	3
B2	SBCA		4	3
B3	SUBD		6	3
B4	ANDA		4	3
B5	BITA		4	3
B6	LDAA		4	3
B7	STAA		4	3
B8	EORA		4	3
B9	ADCA		4	3
BA	ORAA		4	3
BB	ADDA		4	3
BC	CPX		6	3
BD	JSR		6	3
BE	LDS		5	3
BF	STS	EXTND	5	3
C0	SUBB	IMMED	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB		2	2
C6	LDAB		2	2
C7	*			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORAB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	*			
CD	LDX	IMMED	3	3
CF	*			
D0	SUBB	DIR	3	2
D1	CMPB		3	2
D2	SBCB		3	2
D3	ADDD		5	2
D4	ANDB		3	2
D5	BITB		3	2
D6	LDAB		3	2
D7	STAB		3	2
D8	EORB		3	2
D9	ADCB		3	2
DA	ORAB		3	2
DB	ADDB		3	2
DC	LDD		4	2
DD	STD		4	2
DE	LDX		4	2
DF	STX	DIR	4	2
E0	SUBB	INDXD	4	2
E1	CMPB		4	2
E2	SBCB		4	2
E3	ADDD		6	2
E4	ANDB		4	2
E5	BITB		4	2
E6	LDAB		4	2
E7	STAB		4	2
E8	EORB		4	2
E9	ADCB		4	2
EA	ORAB		4	2
EB	ADDB		4	2
EC	LDD		5	2
ED	STD		5	2
EE	LDX		5	2
EF	STX	INDXD	5	2
F0	SUBB	EXTND	4	3
F1	CMPB		4	3
F2	SBCB		4	3
F3	ADDD		6	3
F4	ANDB		4	3
F5	BITB		4	3
F6	LDAB		4	3
F7	STAB		4	3
F8	EORB		4	3
F9	ADCB		4	3
FA	ORAB		4	3
FB	ADDB		4	3
FC	LDD		5	3
FD	STD		5	3
FE	LDX		5	3
FF	STX	EXTND	5	3

*UNDEFINED OP CODE

NOTES:

1. Addressing Modes
   INHER = Inherent INDXD = Indexed IMMED = Immediate
   REL = Relative EXTND = Extended DIR = Direct
2. Unassigned opcodes are indicated by "*" and should not be executed.
3. Codes marked by "T" force the PC to function as a 16-bit counter.

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

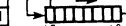
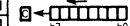
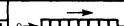
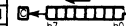
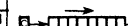
Pointer Operations	Mnemonic	Immed			Direct			Index			Extnd			Inherent			Boolean/ Arithmetic Operation	Condition Codes					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X - M : M + 1	●	●	↕	↕	↕	↕
Decrement Index Reg	DEX													09	3	1	X - 1 → X	●	●	●	↕	●	●
Decrement Stack Pntr	DES													34	3	1	SP - 1 → SP	●	●	●	●	●	●
Increment Index Reg	INX													08	3	1	X + 1 → X	●	●	●	↕	●	●
Increment Stack Pntr	INS													31	3	1	1 SP + 1 → SP	●	●	●	●	●	●
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	●	●	↕	↕	R	●
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	●	●	↕	↕	R	●
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	●	●	↕	↕	R	●
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	●	●	↕	↕	R	●
Index Reg — Stack Pntr	TXS													35	3	1	X - 1 → SP	●	●	●	●	●	●
Stack Pntr — Index Reg	TSX													30	3	1	SP + 1 → X	●	●	●	●	●	●
Add	ABX													3A	3	1	B + X → X	●	●	●	●	●	●
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP - 1 → SP X_H → M_{SP}, SP - 1 → SP	●	●	●	●	●	●
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	●	●	●	●	●	●

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	●	●	●	●	●	●
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	●	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	●	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	●	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	●	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	●	●	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	●	●	↕	↕	R	●
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	●	●	↕	↕	R	●
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	ASLA													48	2	1		●	●	↕	↕	↕	↕
	ASLB													58	2	1		●	●	↕	↕	↕	↕

— Continued —

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS (CONTINUED)

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Shift Left Dbl	ASLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					●	●	↕	↕	↕	↕
	ASRA													47	2	1		●	●	↕	↕	↕	↕
	ASRB													57	2	1		●	●	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	●	●	↕	↕	R	●
Compare Acmltrs	CBA													11	2	1	A - B	●	●	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	●	●	R	S	R	R
	CLRA													4F	2	1	00 → A	●	●	R	S	R	R
	CLRB													5F	2	1	00 → B	●	●	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	●	●	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	●	●	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	●	●	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	●	●	↕	↕	R	S
Decimal Adj. A	DAA													19	2	1	Adj binary sum to BCD	●	●	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	●	●	↕	↕	↕	●
	DECA													4A	2	1	A - 1 → A	●	●	↕	↕	↕	●
	DECB													5A	2	1	B - 1 → B	●	●	↕	↕	↕	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	●	●	↕	↕	↕	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	↕	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	↕	●
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	●	●	↕	↕	R	●
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	●	●	↕	↕	R	●
Logical Shift, Left	LSL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	LSLA													48	2	1		●	●	↕	↕	↕	↕
	LSLB													58	2	1		●	●	↕	↕	↕	↕
	LSLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					●	●	R	↕	↕	↕
	LSRA													44	2	1		●	●	R	↕	↕	↕
	LSRB													54	2	1		●	●	R	↕	↕	↕
	LSRD													04	3	1		●	●	R	↕	↕	↕
Multiply	MUL													3D	10	1	A X B → D	●	●	●	●	●	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	●	●	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	●	●	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	●	●	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	●	●	●	●	●	●
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	3	1	A → Stack	●	●	●	●	●	●
	PSHB													37	3	1	B → Stack	●	●	●	●	●	●
Pull Data	PULA													32	4	1	Stack → A	●	●	●	●	●	●
	PULB													33	4	1	Stack → B	●	●	●	●	●	●
Rotate Left	ROL							69	6	2	79	6	3					●	●	↕	↕	↕	↕
	ROLA													49	2	1		●	●	↕	↕	↕	↕
	ROLB													59	2	1		●	●	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					●	●	↕	↕	↕	↕
	RORA													46	2	1		●	●	↕	↕	↕	↕
	RORB													56	2	1		●	●	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A - B → A	●	●	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	●	●	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	●	●	↕	↕	R	●
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	●	●	↕	↕	R	●
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	●	●	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	●	●	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A - 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B - 00	●	●	↕	↕	R	R

The Condition Code Register notes are listed after Table 12.

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	Mnemonic	Direct OP	Direct ~	Direct #	Relative OP	Relative ~	Relative #	Index OP	Index ~	Index #	Extnd OP	Extnd ~	Extnd #	Inherent OP	Inherent ~	Inherent #	Branch Test	Cond. Code Reg. 5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA				20	3	2										None	●	●	●	●	●	●
Branch Never	BRN				21	3	2										None	●	●	●	●	●	●
Branch If Carry Clear	BCC				24	3	2										C = 0	●	●	●	●	●	●
Branch If Carry Set	BCS				25	3	2										C = 1	●	●	●	●	●	●
Branch If = Zero	BEQ				27	3	2										Z = 1	●	●	●	●	●	●
Branch If ≥ Zero	BGE				2C	3	2										N⊕V = 0	●	●	●	●	●	●
Branch If > Zero	BGT				2E	3	2										Z + (N⊕V) = 0	●	●	●	●	●	●
Branch If Higher	BHI				22	3	2										C + Z = 0	●	●	●	●	●	●
Branch If Higher or Same	BHS				24	3	2										C = 0	●	●	●	●	●	●
Branch If ≤ Zero	BLE				2F	3	2										Z + (N⊕V) = 1	●	●	●	●	●	●
Branch If Carry Set	BLO				25	3	2										C = 1	●	●	●	●	●	●
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	●	●	●	●	●	●
Branch If < Zero	BLT				2D	3	2										N⊕V = 1	●	●	●	●	●	●
Branch If Minus	BMI				2B	3	2										N = 1	●	●	●	●	●	●
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	●	●	●	●	●	●
Branch If Overflow Clear	BVC				28	3	2										V = 0	●	●	●	●	●	●
Branch If Overflow Set	BVS				29	3	2										V = 1	●	●	●	●	●	●
Branch If Plus	BPL				2A	3	2										N = 0	●	●	●	●	●	●
Branch To Subroutine	BSR				8D	6	2										See Special Operations - Figure 26	●	●	●	●	●	●
Jump	JMP							6E	3	2	7E	3	3					●	●	●	●	●	●
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					●	●	●	●	●	●
No Operation	NOP													01	2	1		●	●	●	●	●	●
Return From Interrupt	RTI													3B	10	1	See Special Operations - Figure 26	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait For Interrupt	WAI													3E	9	1		●	●	●	●	●	●

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

Operations	Inherent Mnemonic	OP	~	#	Boolean Operation	Cond. Code Reg. 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	0 → C	●	●	●	●	●	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	●	R	●	●	●	●
Clear Overflow	CLV	0A	2	1	0 → V	●	●	●	●	R	●
Set Carry	SEC	0D	2	1	1 → C	●	●	●	●	●	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	●	S	●	●	●	●
Set Overflow	SEV	0B	2	1	1 → V	●	●	●	●	S	●
Accumulator A → CCR	TAP	06	2	1	A → CCR	↕	↕	↕	↕	↕	↕
CCR → Accumulator A	TPA	07	2	1	CCR → A	●	●	●	●	●	●

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MPU Cycles
MSP Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
- Arithmetic Minus
● Boolean AND
X Arithmetic Multiply
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
M̄ Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
↕ Affected
● Not Affected

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E-CYCLES

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
ABA	•	•	•	•	2	•
ABX	•	•	•	•	3	•
ADC	2	3	4	4	•	•
ADD	2	3	4	4	•	•
ADDD	4	5	6	6	•	•
AND	2	3	4	4	•	•
ASL	•	•	6	6	2	•
ASLD	•	•	•	•	3	•
ASR	•	•	6	6	2	•
BCC	•	•	•	•	•	3
BCS	•	•	•	•	•	3
BEQ	•	•	•	•	•	3
BGE	•	•	•	•	•	3
BGT	•	•	•	•	•	3
BHI	•	•	•	•	•	3
BHS	•	•	•	•	•	3
BIT	2	3	4	4	•	•
BLE	•	•	•	•	•	3
BLO	•	•	•	•	•	3
BLS	•	•	•	•	•	3
BLT	•	•	•	•	•	3
BMI	•	•	•	•	•	3
BNE	•	•	•	•	•	3
BPL	•	•	•	•	•	3
BRA	•	•	•	•	•	3
BRN	•	•	•	•	•	3
BSR	•	•	•	•	•	6
BVC	•	•	•	•	•	3
BVS	•	•	•	•	•	3
CBA	•	•	•	•	2	•
CLC	•	•	•	•	2	•
CLI	•	•	•	•	2	•
CLR	•	•	6	6	2	•
CLV	•	•	•	•	2	•
CMP	2	3	4	4	•	•
COM	•	•	6	6	2	•
CPX	4	5	6	6	•	•
DAA	•	•	•	•	2	•
DEC	•	•	6	6	2	•
DES	•	•	•	•	3	•
DEX	•	•	•	•	3	•
EOR	2	3	4	4	•	•
INC	•	•	6	6	•	•
INS	•	•	•	•	3	•

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
INX	•	•	•	•	3	•
JMP	•	•	3	3	•	•
JSR	•	5	6	6	•	•
LDA	2	3	4	4	•	•
LDD	3	4	5	5	•	•
LDS	3	4	5	5	•	•
LDX	3	4	5	5	•	•
LSL	•	•	6	6	2	•
LSLD	•	•	•	•	3	•
LSR	•	•	6	6	2	•
LSRD	•	•	•	•	3	•
MUL	•	•	•	•	10	•
NEG	•	•	6	6	2	•
NOP	•	•	•	•	2	•
ORA	2	3	4	4	•	•
PSH	•	•	•	•	3	•
PSHX	•	•	•	•	4	•
PUL	•	•	•	•	4	•
PULX	•	•	•	•	5	•
ROL	•	•	6	6	2	•
ROR	•	•	6	6	2	•
RTI	•	•	•	•	10	•
RTS	•	•	•	•	5	•
SBA	•	•	•	•	2	•
SBC	2	3	4	4	•	•
SEC	•	•	•	•	2	•
SEI	•	•	•	•	2	•
SEV	•	•	•	•	2	•
STA	•	3	4	4	•	•
STD	•	4	5	5	•	•
STS	•	4	5	5	•	•
STX	•	4	5	5	•	•
SUB	2	3	4	4	•	•
SUBD	4	5	6	6	•	•
SWI	•	•	•	•	12	•
TAB	•	•	•	•	2	•
TAP	•	•	•	•	2	•
TBA	•	•	•	•	2	•
TPA	•	•	•	•	2	•
TST	•	•	6	6	2	•
TSX	•	•	•	•	3	•
TXS	•	•	•	•	3	•
WAI	•	•	•	•	9	•

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 14 provides a detailed description of the information present on the address bus, data bus, and the read/write (R/$\overline{W}$) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Note that during MPU reads of internal locations, the resultant value will not appear on the external data bus except in mode 0. "High order" byte refers to the most-significant byte of a 16-bit value.

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 1 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Operand Data
LDS LDX LDD	3	1 2 3	Opcode Address Opcode Address + 1 Opcode Address + 2	1 1 1	Opcode Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Opcode Address + 2 Address Bus FFFF	1 1 1 1	Opcode Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
DIRECT					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Opcode Address Opcode Address + 1 Address of Operand	1 1 1	Opcode Address of Operand Operand Data
STA	3	1 2 3	Opcode Address Opcode Address + 1 Destination Address	1 1 0	Opcode Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Opcode Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Opcode Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Opcode Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Subroutine Address Stack Pointer Stack Pointer − 1	1 1 1 0 0	Opcode Irrelevant Data First Subroutine Opcode Return Address (Low Order Byte) Return Address (High Order Byte)

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 2 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Jump Address (High Order Byte)
		3	Opcode Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Address of Operand
AND ORA		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Destination Address (High Order Byte)
		3	Opcode Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
STD		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Opcode Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Subroutine (High Order Byte)
		3	Opcode Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

*TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 3 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

*TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

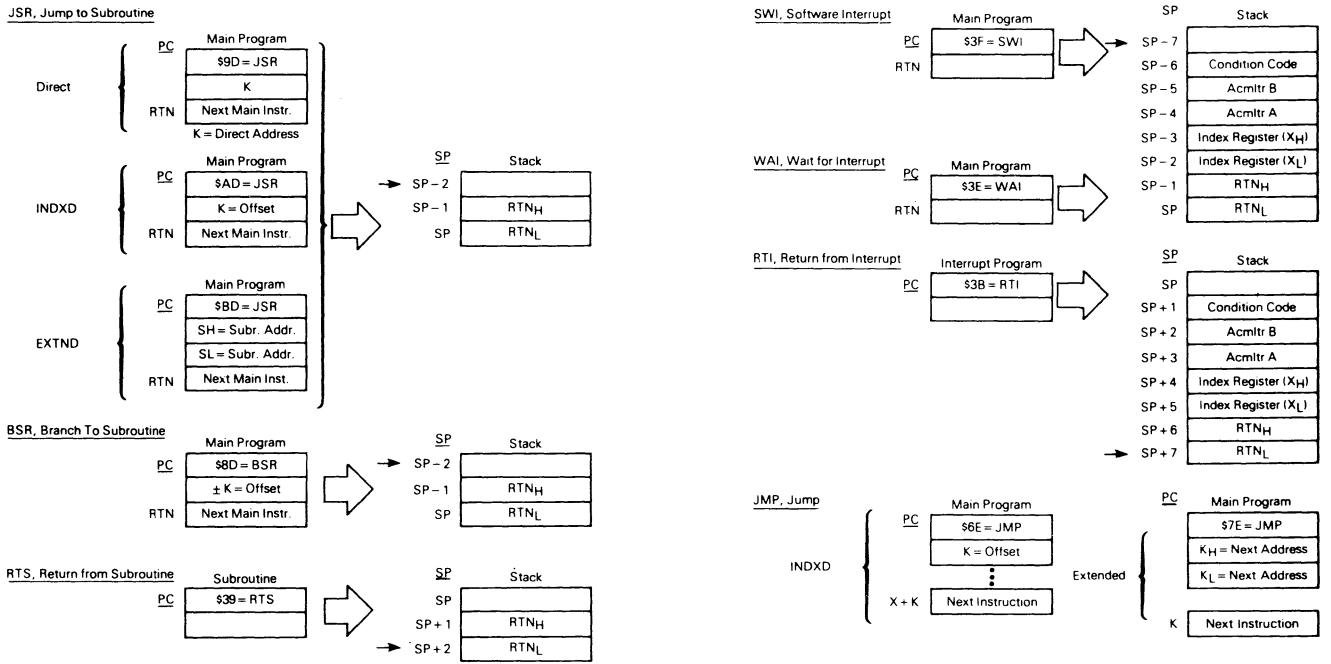
TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Opcode Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer – 1 Stack Pointer – 2 Stack Pointer – 3 Stack Pointer – 4 Stack Pointer – 5 Stack Pointer – 6	1 1 0 0 0 0 0 0 0	Opcode Opcode of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Condition Code Register

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 5 of 5)

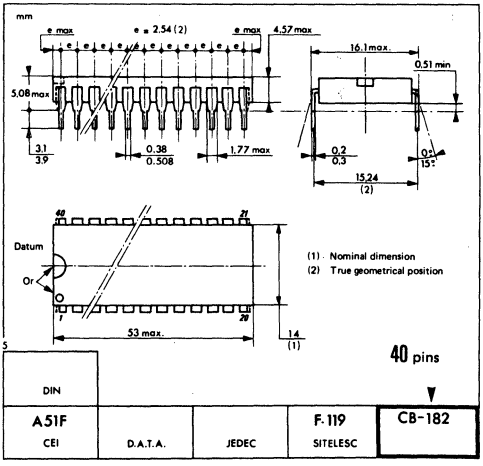
Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
MUL	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer − 1	0	Return Address (High Order Byte)
		5	Stack Pointer − 2	0	Index Register (Low Order Byte)
		6	Stack Pointer − 3	0	Index Register (High Order Byte)
		7	Stack Pointer − 4	0	Contents of Accumulator A
		8	Stack Pointer − 5	0	Contents of Accumulator B
		9	Stack Pointer − 6	0	Contents of Condition Code Register
		10	Stack Pointer − 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMI BVS	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Buss FFFF	1	Low Byte of Restart Vector
BSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

FIGURE 25 – SPECIAL OPERATIONS

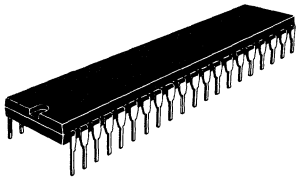


Legend:
RTN = Address of next instruction in Main Program to be executed upon return from subroutine
RTN$_H$ = Most significant byte of Return Address
RTN$_L$ = Least significant byte of Return Address
→ = Stack Pointer After Execution
K = 8-bit Unsigned Value

PHYSICAL DIMENSIONS



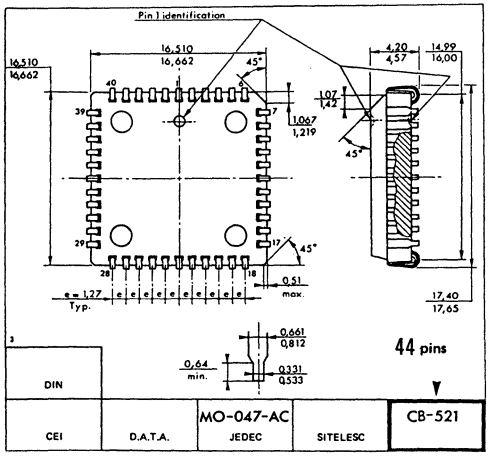
CB-182



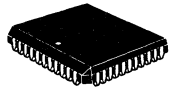
P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

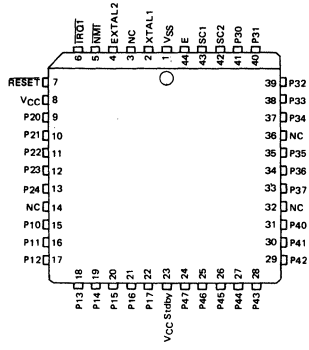
C SUFFIX
CERAMIC PACKAGE



CB-521



FN SUFFIX
PLCC 44



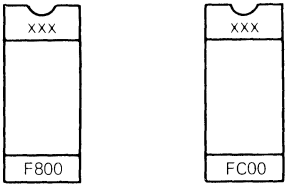
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to THOMSON SEMICONDUCTEURS on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local THOMSON SEMICONDUCTEURS representative or distributor.

EPROMs

Two 2708 or one 2716 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



XXX = Customer ID

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to THOMSON SEMICONDUCTEURS. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, THOMSON SEMICONDUCTEURS will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by THOMSON SEMICONDUCTEURS. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6801 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from THOMSON SEMICONDUCTEURS factory representatives.

EFDOS is THOMSON SEMICONDUCTEURS' Disk Operating System available on development systems such as DEVICE,...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser,...

*Requires prior factory approval.

Whenever ordering a custom MCU is required, please contact your local THOMSON SEMICONDUCTEURS representative or THOMSON SEMICONDUCTEURS distributor and/or complete and send the attached "MCU customer ordering sheet" to your local THOMSON SEMICONDUCTEURS representative.

ORDERING INFORMATION

EF6803 P V

Device — Package — Oper. temp. — Screening level

The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

DEVICE		PACKAGE					OPER. TEMP.			SCREENING LEVEL			
		C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
1.0 MHz	EF6801/03			●		●	●			●			
				●				●		●			
	EF6803	●						●		●			
1.25 MHz	EF6801/03-1			●		●	●			●			
				●				●		●			
	EF6803-1	●						●		●			
1.5 MHz	EF68A01/03			●			●			●			
	EF68A03	●						●		●			
2.0 MHz	EF68B01/03			●			●			●			

Examples : EF6801P, EF6801FN, EF6801PV, EF6803CV

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** −40°C to +85°C, **M :** −55°C to +125°C, ***** : may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.

EXORciser is a registered trade mark of MOTOROLA Inc.

THOMSON COMPONENTS
MOSTEK

EF6801U4 • EF6803U4

MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

ADVANCE INFORMATION

The EF6801U4 is an 8-bit single-chip microcomputer unit (MCU) which enhances the capabilities of the EF6801 and significantly enhances the capabilities of the EF6800 Family of parts. It includes an EF6801 microprocessor unit (MPU) with direct object-code compatibility and upward object-code compatibility with the EF6800. Execution times of key instructions have been improved over the EF6800 and the new instructions found on the EF6801 are included. The MCU can function as a monolithic microcomputer or can be expanded to a 64K byte address space. It is TTL compatible and requires one +5-volt power supply. On-chip resources include 4096 bytes of ROM, 192 bytes of RAM, a serial communications interface (SCI), parallel I/O, and a 16-bit six-function programmable timer. The EF6803U4 can be considered as an EF6801U4 operating in modes 2 or 3 ; i.e., those that do not use internal ROM.

- Enhanced EF6800 Instruction Set
- Upward Source and Object Code Compatibility with the EF6800 and EF6801
- Bus Compatibility with the EF6800 Family
- 8 x 8 Multiply Instruction
- Single-Chip or Expanded Operation to 64K Byte Address Space
- Internal Clock Generator with Divide-by-Four Output
- Serial Communications Interface (SCI)
- 16-Bit Six-Function Programmable Timer
- Three Output Compare Functions
- Two Input Capture Functions
- Counter Alternate Address
- 4096 Bytes of ROM (EF6801U4)
- 192 Bytes of RAM
- 32 Bytes of RAM Retainable During Power Down
- 29 Parallel I/O and Two Handshake Control Lines
- $\overline{NMI}$ Inhibited Until Stack Load
- Complete Development System Support on DEVICE®
- −40°C to 85°C Temperature Range

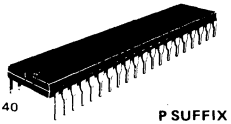
DEVICE® is THOMSON SEMICONDUCTEURS' development/emulation tool.

HMOS
HIGH-DENSITY
N-CHANNEL, SILICON-GATE

MICROCOMPUTER/
MICROPROCESSOR

CASES

CB-182

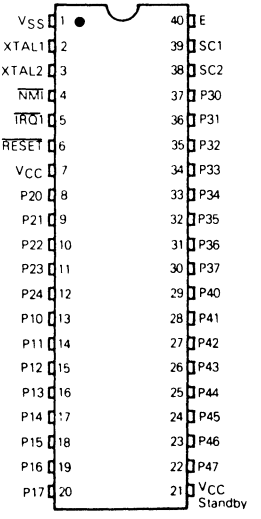


P SUFFIX
PLASTIC PACKAGE

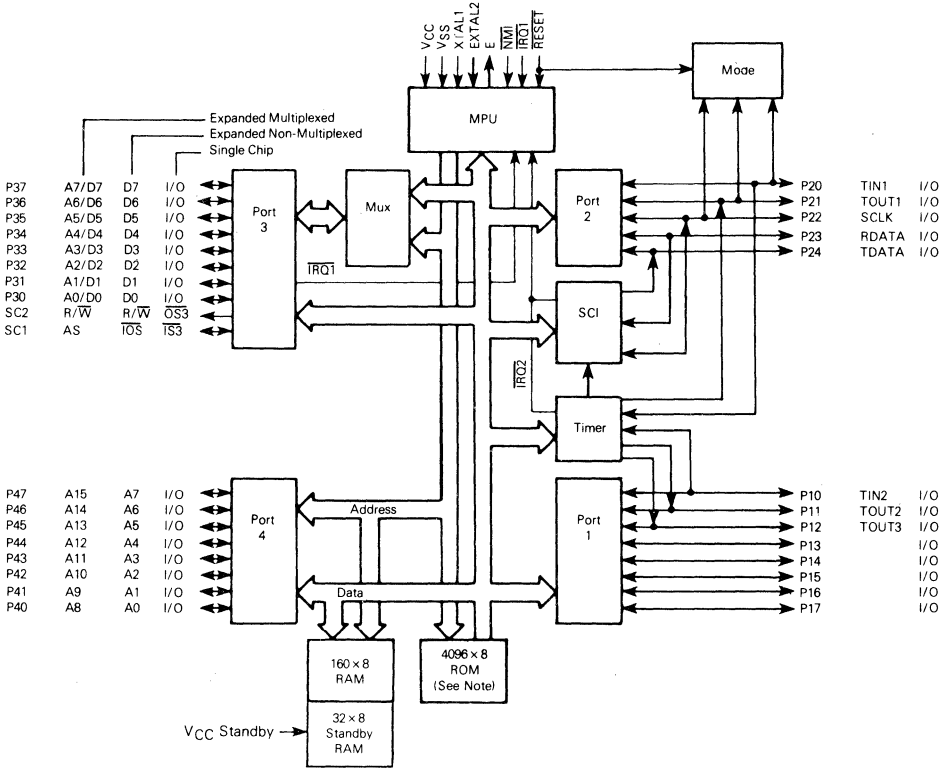
CB-521

FN SUFFIX
PLCC 44

PIN ASSIGNMENT



EF6801U4 MICROCOMPUTER FAMILY BLOCK DIAGRAM



NOTE : No functioning ROM in EF6803U4.

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range EF6801/03U4, EF6801/03U4-1, EF68A01/03U4 EF6801/03U4, EF6801/03U4-1 : V suffix	T_A	T_H to T_L 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$. Input protection is enhanced by connecting unused inputs to either V_{DD} or V_{SS}.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance Plastic PLCC	θ_{JA}	 50 100	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

CONTROL TIMING (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = 0 to 70°C)

Characteristic	Symbol	EF6801U4 EF6803U4		EF6801U4-1 EF6803U4-1		EF68A01U4 EF68A03U4		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	0.5	1.0	0.5	1.25	0.5	1.5	MHz
Crystal Frequency	f_{XTAL}	2.0	4.0	2.0	5.0	2.0	6.0	MHz
External Oscillator Frequency	$4 f_o$	2.0	4.0	2.0	5.0	2.0	6.0	MHz
Crystal Oscillator Startup Time	t_{rc}	—	100	—	100	—	100	ms
Processor Control Setup Time	t_{PCS}	200	—	170	—	140	—	ns

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	EF6801/03U4-A01/03U4 0° to +70°C		EF6801U4/6803U4 −40° to +85°C		Unit
			Min	Max	Min	Max	
Input High Voltage	$\overline{RESET}$	V_{IH}	$V_{SS}+4.0$	V_{CC}	$V_{SS}+4.0$	V_{CC}	V
	Other Inputs*		$V_{SS}+2.0$	V_{CC}	$V_{SS}+2.2$	V_{CC}	
Input Low Voltage	All Inputs*	V_{IL}	$V_{SS}-0.3$	$V_{SS}+0.8$	$V_{SS}-0.3$	$V_{SS}+0.8$	V
Input Load Current (V_{in} = 0 to 2.4 V)	Port 4	I_{in}	–	0.5	–	0.8	mA
	SCI		–	0.8	–	1.0	
Input Leakage Current (V_{in} = 0 to 5.5 V)	$\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}	–	2.5	–	5.0	μA
Hi-Z (Off-State) Input Current (V_{in} = 0.5 to 2.4 V)	Port 1, Port 2, Port 3	I_{TSI}	–	10	–	20	μA
Output High Voltage (I_{Load} = −65 μA, V_{CC} = Min)	Port 4, SC1, SC2	V_{OH}	$V_{SS}+2.4$	–	$V_{SS}+2.4$	–	V
(I_{Load} = −100 μA, V_{CC} = Min)	Other Outputs		$V_{SS}+2.4$	–	$V_{SS}+2.4$	–	
Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = Min)	All Outputs	V_{OL}	–	$V_{SS}+0.5$	–	$V_{SS}+0.6$	V
Darlington Drive Current (V_O = 1.5 V)	Port 1	I_{OH}	1.0	4.0	1.0	5.0	mA
Internal Power Dissipation (Measured at $T_A = T_L$ in Steady-State Operation)***		P_{INT}	–	1200	–	1500	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f_o = 1.0 MHz)	Port 3, Port 4, SC1	C_{in}	–	12.5	–	12.5	pF
	Other Inputs		–	10.0	–	10.0	
V_{CC} Standby	Powerdown	V_{SBB}	4.0	5.25	4.0	5.25	V
	Powerup	V_{SB}	4.75	5.25	4.75	5.25	
Standby Current	Powerdown	I_{SBB}	–	3.0	–	3.5	mA

* Except mode programming levels; see Figure 16.

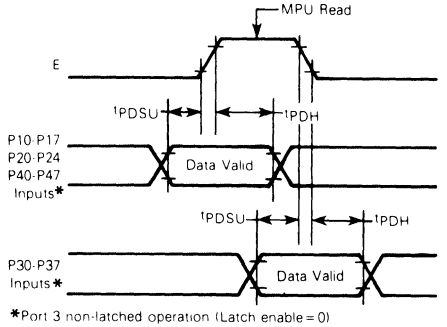
** Negotiable to −100 μA (for further information contact the factory).

*** For the EF6801U4/EF6803U4 T_L = 0°C and for the EF6801U4/EF6803U4 : V suffix T_L = −40°C

PERIPHERAL PORT TIMING (Refer to Figures 1-4)

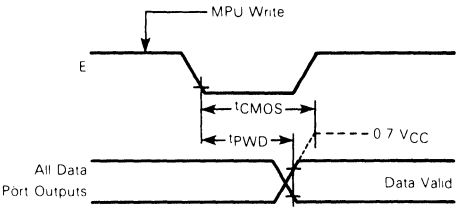
Characteristic	Symbol	EF6801/03U4 EF6801/03U4-1			EF68A01/03U4			Unit
		Min	Typ	Max	Min	Typ	Max	
Peripheral Data Setup Time	t_{PDSU}	200	–	–	150	–	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	–	150	–	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	–	350	–	–	300	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	–	350	–	–	300	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid Port 1	t_{PWD}	–	–	350	–	–	300	ns
Port 2, 3, 4		–	–	350	–	–	300	
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	–	2.0	–	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	–	150	–	–	ns
Input Data Hold Time	t_{IH}	50	–	–	40	–	–	ns
Input Data Setup Time	t_{IS}	20	–	–	20	–	–	ns

FIGURE 1 – DATA SETUP AND HOLD TIMES
(MPU READ)



*Port 3 non-latched operation (Latch enable = 0)

FIGURE 2 – DATA SETUP AND HOLD TIMES
(MPU WRITE)



NOTES
1. 10 k pullup resistor required for port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

FIGURE 3 - PORT 3 OUTPUT STROBE TIMING
(EF6801U4 SINGLE-CHIP MODE)

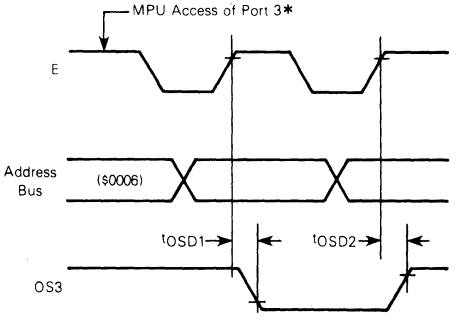
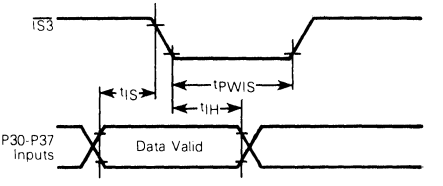


FIGURE 4 - PORT 3 LATCH TIMING
(EF6801U4 SINGLE-CHIP MODE)



*Access matches output strobe select (OSS = 0, a read; OSS = 1, a write)

NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 5 – CMOS LOAD

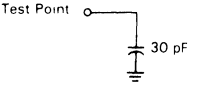
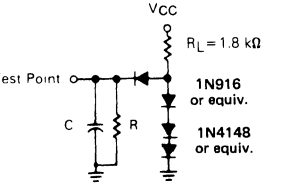


FIGURE 6 – TIMING TEST
LOAD PORTS 1, 2, 3, AND 4



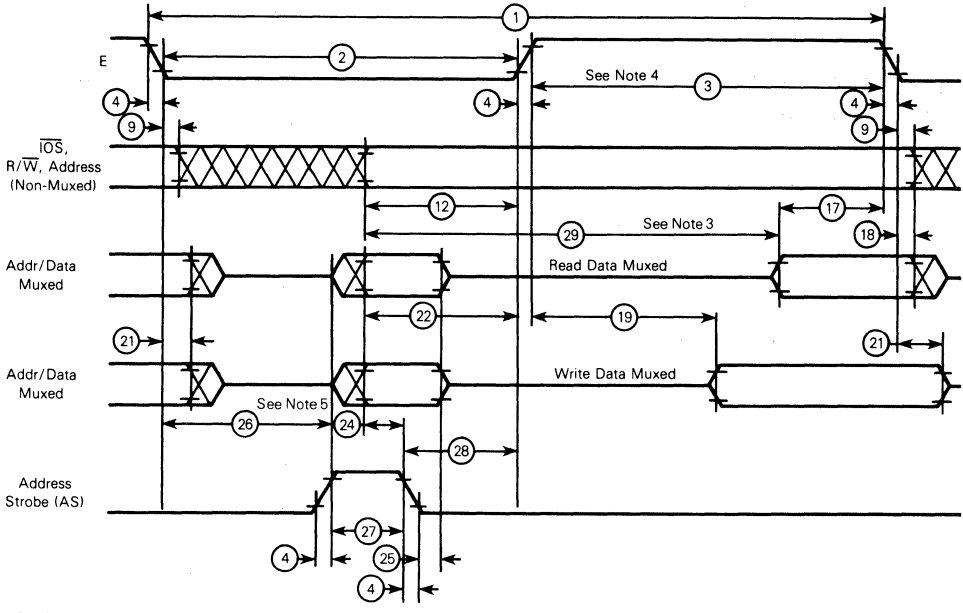
C = 90 pF for P30-P37, P40-P47, E, SC1, SC2
= 30 pF for P10-P17, P20-P24
R = 37 kΩ for P40-P47, SC1, SC2
= 24 kΩ for P10-P17, P20-P24
= 24 kΩ for P30-P37, E

BUS TIMING (See Notes 1 and 2, and Figure 7)

Ident. Number	Characteristics	Symbol	EF6801U4 EF6803 U4		EF6801U4-1 EF6803U4-1		EF68A01U4 EF68A03U4		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	0.66	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	300	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	300	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	25	ns
9	Address Hold Time	t_{AH}	20	–	20	–	20	–	ns
12	Non-Muxed Address Valid Time to E*	t_{AV}	200	–	150	–	115	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	60	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	–	160	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	20	–	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	160	–	120	–	100	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	40	–	30	–	30	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	20	–	20	–	ns
26	Delay Time, E to AS Rise*	t_{ASD}	200	–	170	–	130	–	ns
27	Pulse Width, AS High*	PW_{ASH}	100	–	80	–	60	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	90	–	70	–	60	–	ns
29	Usable Access Time* (See Note 3)	t_{ACC}	555	–	435	–	385	–	ns

*At specified cycle time.

FIGURE 7 – BUS TIMING



NOTES:
1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Usable access time is computed by 22 + 3 – 17 + 4.
4. Memory devices should be enabled only during E high to avoid port 3 bus contention.
5. Item 26 is different from the EF6801 but it is upward compatible.

INTRODUCTION

The EF6801U4 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a data register and a write-only data direction register. The data direction register is used to define whether corresponding bits in the data register are configured as an input (clear) or output (set).

The term "port" by itself refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port," it is controlled by the port data direction register and the programmer has direct access to the port pins using the port data register. Port pins are labeled as Pij where i identifies one of four ports and j indicates the particular bit.

The microprocessor unit (MPU) is an enhanced EF6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the EF6800 and the EF6801. The programming model is depicted in Figure 8 where accumulator D is a concatenation of accumulators A and B. A list of new operations added to the EF6800 instruction set are shown in Table 1.

The EF6803U4 can be considered an EF6801U4 that operates in modes 2 and 3 only.

FIGURE 8 — PROGRAMMING MODEL

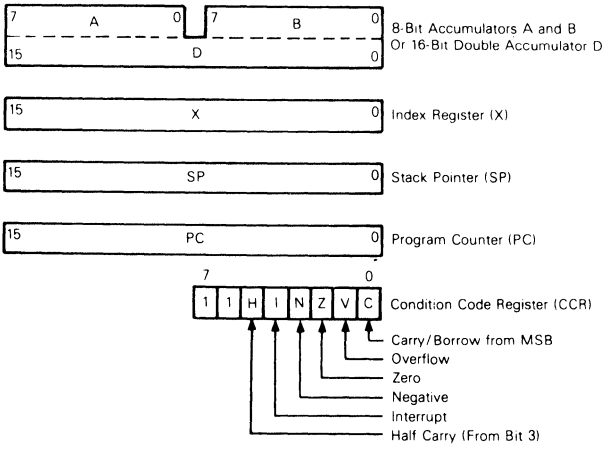


TABLE 1 — NEW INSTRUCTIONS

Instruction	Description
ABX	Unsigned addition of accumulator B to index register
ADDD	Adds (without carry) the double accumulator to memory and leaves the sum in the double accumulator
ASLD or LSLD	Shifts the double accumulator left (towards MSB) one bit, the LSB is cleared, and the MSB is shifted into the C bit
BHS	Branch if higher or same, unsigned conditional branch (same as BCC)
BLO	Branch if lower, unsigned conditional branch (same as BCS)
BRN	Branch never
JSR	Additional addressing mode direct
LDD	Loads double accumulator from memory
LSL	Shifts memory or accumulator left (towards MSB) one bit, the LSB is cleared, and the MSB is shifted into the C bit (same as ASL)
LSRD	Shifts the double accumulator right (towards LSB) one bit, the MSB is cleared, and the LSB is shifted into the C bit
MUL	Unsigned multiply, multiplies the two accumulators and leaves the product in the double accumulator
PSHX	Pushes the index register to stack
PULX	Pulls the index register from stack
STD	Stores the double accumulator to memory
SUBD	Subtracts memory from the double accumulator and leaves the difference in the double accumulator
CPX	Internal processing modified to permit its use with any conditional branch instruction

OPERATING MODES

The EF6801U4 provides seven different operating modes (modes 0 through 3 and 5 through 7) and the EF6803U4 provides two operating modes (modes 2 and 3). The operating modes are hardware selectable and determine the device memory map, the configuration of port 3, port 4, SC1, SC2, and the physical location of the interrupt vectors.

FUNDAMENTAL MODES

The seven operating modes (0-3, 5-7) can be grouped into three fundamental modes which refer to the type of bus it supports: single chip, expanded non-multiplexed, and expanded multiplexed. Single chip is mode 7, expanded non-multiplexed is mode 5, and the remaining 5 are expanded multiplexed modes. Table 2 summarizes the characteristics of the operating modes.

EF6801U4 SINGLE-CHIP MODE (7) -- In the single-chip mode, the four MCU ports are configured as parallel input/output data ports, as shown in Figure 9. The MCU functions as a monolithic microcomputer in this mode without external address or data buses. A maximum of 29 I/O lines and two port 3 control lines are provided. Peripherals or another MCU can be interfaced to port 3 in a loosely coupled dual-processor configuration, as shown in Figure 10.

TABLE 2 – SUMMARY OF EF6801U4/EF6803U4 OPERATING MODES

Single-Chip (Mode 7)
- 192 bytes of RAM, 4096 bytes of ROM
- Port 3 is a parallel I/O port with two control lines
- Port 4 is a parallel I/O port

Expanded Non-Multiplexed (Mode 5)
- 192 bytes of RAM, 4096 bytes of ROM
- 256 bytes of external memory space
- Port 3 is an 8-bit data bus
- Port 4 is an input port/address bus

Expanded Multiplexed (Modes 0, 1, 2, 3, 6*)
- Four memory space options (total 64K address space)
  - (1) Internal RAM and ROM with partial address bus (mode 1)
  - (2) Internal RAM, no ROM (mode 2)
  - (3) Extended addressing of internal I/O and RAM
  - (4) Internal RAM and ROM with partial address bus (mode 6)
- Port 3 is multiplexed address/data bus
- Port 4 is address bus (inputs/address in mode 6)
- Test mode (mode 0):
  - May be used to test internal RAM and ROM
  - May be used to test ports 3 and 4 as I/O ports by writing into mode 7
- Only modes 5, 6, and 7 can be irreversibly entered from mode 0

Resources Common to All Modes
- Reserved register area
- Port 1 input/output operation
- Port 2 input/output operation
- Timer operation
- Serial communications interface operation

***The EF6803U4 operates only in modes 2 and 3.**

FIGURE 9 — SINGLE-CHIP MODE

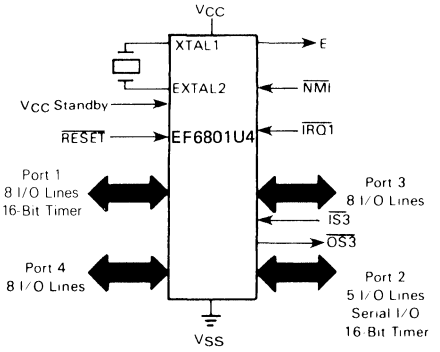
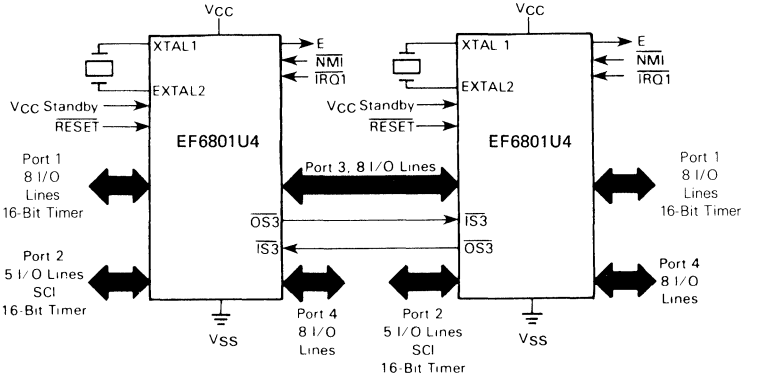


FIGURE 10 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION



EF6801U4 EXPANDED NON-MULTIPLEXED MODE (5) — A modest amount of external memory space is provided in the expanded non-multiplexed mode while significant on-chip resources are retained. Port 3 functions as an 8-bit bidirectional data bus and port 4 is configured initially as an input data port. Any combination of the eight least significant address lines may be obtained by writing to the port 4 data direction register. Stated alternatively, any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Internal pullup resistors pull the port 4 lines high until the port is configured.

Figure 11 illustrates a typical system configuration in the expanded non-multiplexed mode. The MCU interfaces directly with EF6800 family parts and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and can be used as a memory-page select or chip-select line.

EXPANDED MULTIPLEXED MODES (0, 1, 2, 3, 6) — A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes, port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of address strobe (AS) and data valid while E is high. In modes 0, 2, and 3, port 4 provides address lines A8 to A15. In modes 1 and 6, however, port 4 initially is configured at reset as an input data port. The port 4 data direction register can then be changed to provide any combination of address lines A8 to A15. Stated alternatively, any subset of A8 to A15 can be provided while retaining the remaining port 4 lines as input data lines. Internal pullup resistors pull the port 4 lines high until software configures the port. In mode 1, the internal pullup resistors will hold the upper address lines high producing a value of $FFXX for a reset vector. A simple method of getting the desired address lines configured as outputs is to have an external EPROM not fully decoded so it appears at

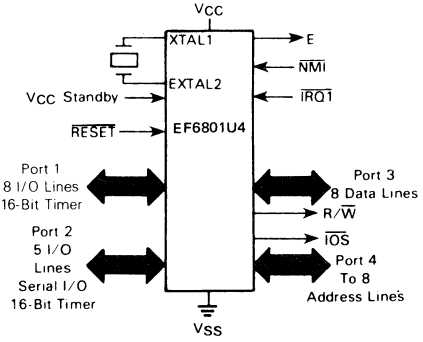
two address locations (i.e., $FXXX and $BXXX). Then, when the reset vector appears as $FFFE, the EPROM will be accessed and can point to an address in the top $100 bytes of the internal or external ROM/EPROM that will configure port 4 as desired.

In mode 0, the reset and interrupt vectors are located at $BFF0-$BFFF. In addition, the internal and external data buses are connected, so there must be no memory map overlap in order to avoid potential bus conflicts. By writing the PC0-PC2 bits in the port 2 data register, modes 5, 6, and 7 can be irreversibly entered from mode 0. Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with the automated test equipment.

Only the EF6801U4 can operate in each of the expanded multiplexed modes. The EF6803U4 operates only in modes 2 and 3.

Figure 12 depicts a typical configuration for the expanded multiplexed modes. Address strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in Figure 13. This allows port 3 to function as a data bus when E is high.

FIGURE 11 — EXPANDED NON-MULTIPLEXED CONFIGURATION



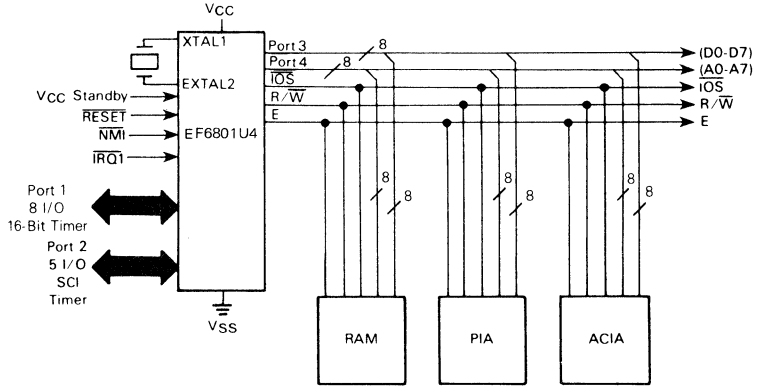
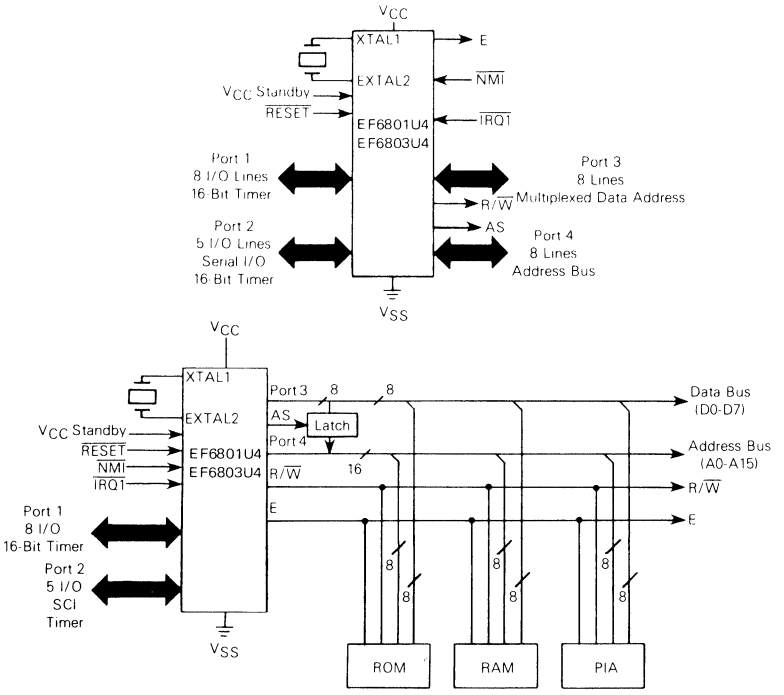
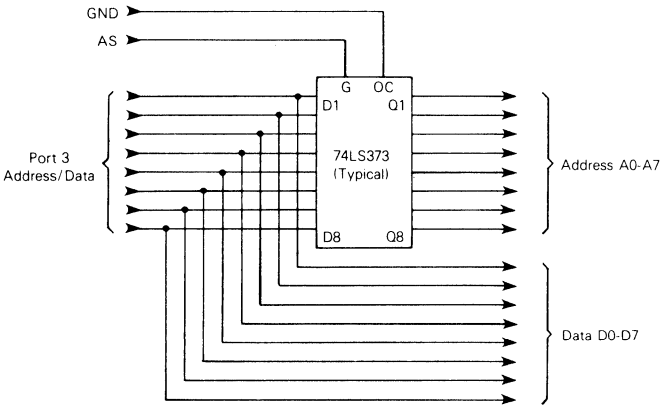


FIGURE 12 – EXPANDED MULTIPLEXED CONFIGURATION



NOTE: To avoid data bus (port 3) contention in the expanded multiplexed modes, memory devices should be enabled only during E high time

FIGURE 13 – TYPICAL LATCH ARRANGEMENT



PROGRAMMING THE MODE

The operating mode is determined at $\overline{RESET}$ by the levels asserted on P22, P21, and P20. These levels are latched into PC2, PC1, and PC0 of the program control register on the positive edge of $\overline{RESET}$. The operating mode may be read from the port 2 data register, as shown below, and programming levels and timing must be met as shown in Figure 14. A brief outline of the operating modes is shown in Table 3.

PORT 2 DATA REGISTER

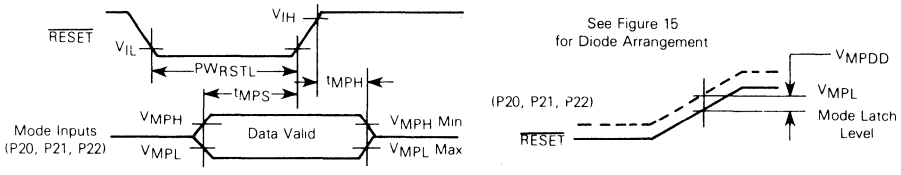
7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in Figure 15 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

The EF6801U4/EF6803U4 can provide up to 64K byte address space depending on the operating mode. A memory map for each operating mode is shown in Figure 16. The first 32 locations of each map are reserved for the internal register area, as shown in Table 4, with exceptions as indicated.

FIGURE 14 – MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 14)

Characteristic	Symbol	Min	Max	Unit
Mode Programming Input Voltage Low	V_{MPL}	–	1.8	V
Mode Programming Input Voltage High	V_{MPH}	4.0	–	V
Mode Programming Diode Differential (If Diodes are Used)	V_{MPDD}	0.6	–	V
$\overline{RESET}$ Low Pulse Width	PW_{RSTL}	3.0	–	E Cycles
Mode Programming Setup Time	t_{MPS}	2.0	–	E Cycles
Mode Programming Hold Time $\overline{RESET}$ Rise Time ≥ 1 μs $\overline{RESET}$ Rise Time < 1 μs	t_{MPH}	0 100	– –	ns

TABLE 3 – MODE SELECTION SUMMARY

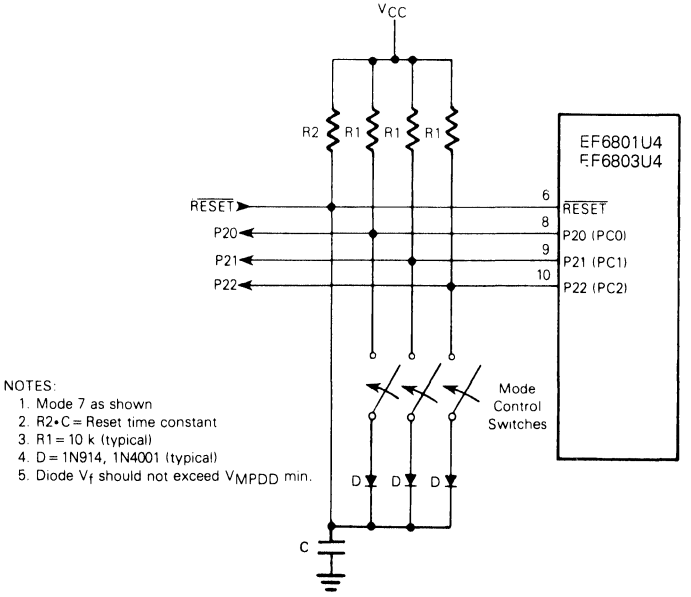
Mode*	P22 PC2	P21 PC1	P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(2, 3)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(2, 3)	Non-Multiplexed/Partial Decode
4	H	L	L	–	–	–	–	Undefined(4)
3	L	H	H	E	I	E	MUX(1, 5)	Multiplexed/RAM
2	L	H	L	E	I	E	MUX(1)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX(1, 3)	Multiplexed/RAM and ROM
0	L	L	L	I	I	E	MUX(1)	Multiplexed Test

LEGEND
I – Internal
E – External
MUX – Multiplexed
NMUX – Non-Multiplexed
L – Logic "0"
H – Logic "1"

NOTES:
1. Addresses associated with ports 3 and 4 are considered external in modes 0, 1, 2, and 3.
2. Addresses associated with port 3 are considered external in modes 5 and 6.
3. Port 4 default is user data input; address output is optional by writing to port 4 data direction register.
4. Mode 4 is a non-user mode and should not be used as an operating mode.
5. Mode 3 has the internal RAM and internal registers relocated at $D000-$D0FF.

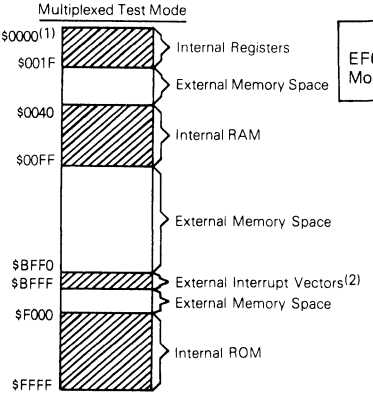
* The EF6803U4 operates only in modes 2 and 3.

FIGURE 15 – TYPICAL MODE PROGRAMMING CIRCUIT



NOTES:
1. Mode 7 as shown
2. R2•C = Reset time constant
3. R1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min.

FIGURE 16 – EF6801U4/EF6803U4 MEMORY MAPS (Sheet 1 of 4)

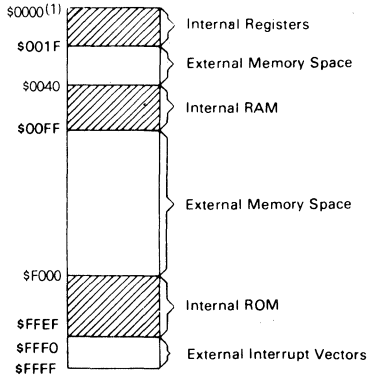


NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) The interrupt vectors are at $BFF0-$BFFF.
3) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which may be used to examine the entire ROM using an external $\overline{\text{RESET}}$ vector.
5) Modes 5-7 can be irreversibly entered from mode 0 by writing to the PC0-PC2 bits of the port 2 data register.

FIGURE 16 – EF6801U4/EF6803U4 MEMORY MAPS (Sheet 2 of 4)

EF6801U4 Mode **1**

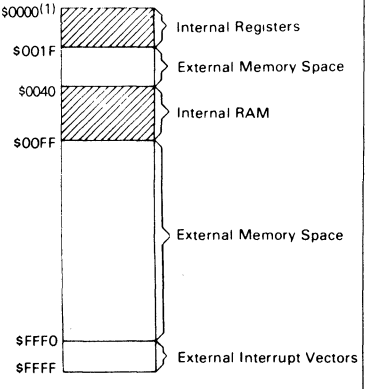
Multiplexed/RAM & ROM



NOTES:
1) Excludes the following addresses which may be used externally: $05 and $07.
2) Internal ROM addresses $FFF0 to $FFFF are not usable.
3) Address lines A8-A15 will not contain addresses until the data direction register for port 4 has been written with "1s" in the appropriate bits. These address lines will assert "1s" until made outputs by writing the data direction register.

EF6801U4 EF6803U4 Mode **2**

Multiplexed/RAM

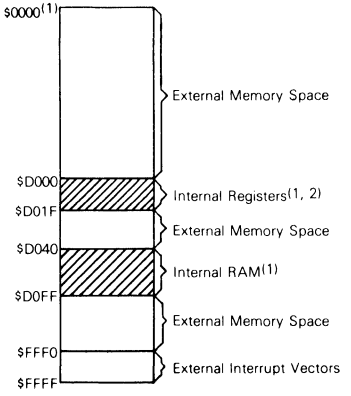


NOTE:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

FIGURE 16 – EF6801U4/EF6803U4 MEMORY MAPS (Sheet 3 of 4)

EF6801U4
EF6803U4
Mode **3**

Multiplexed/RAM

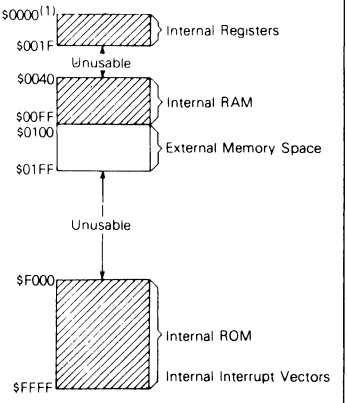


NOTES:

1) Relocating the internal registers and the internal RAM to high memory allows the processor to make use of direct addressing.
2) Excludes the following addresses which may be used externally: $D004, $D005, $D006, $D007, and $D00F.

EF6801U4
Mode **5**

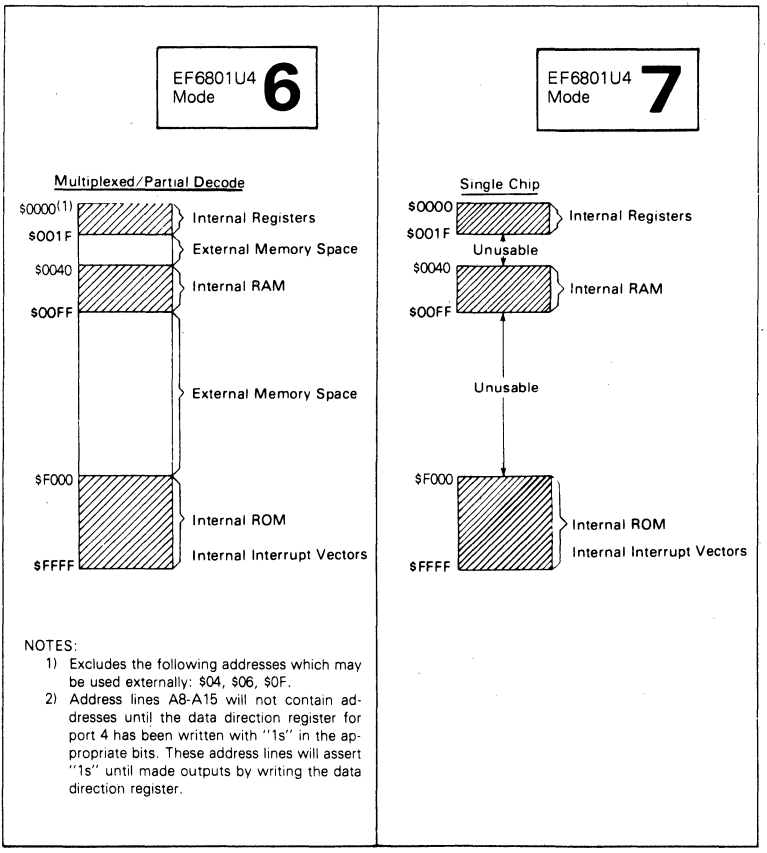
Non-Multiplexed/Partial Decode



NOTES:

1) Excludes the following addresses which may not be used externally: $04, $06, and $0F (no $\overline{IOS}$).
2) Address lines A0 to A7 will not contain addresses until the data direction register for port 4 has been written with "1s" in the appropriate bits. These address lines will assert "1s" until made outputs by writing the data direction register.

FIGURE 16 – EF6801U4/EF6803U4 MEMORY MAPS (Sheet 4 of 4)



NOTES:

1) Excludes the following addresses which may be used externally: $04, $06, $0F.
2) Address lines A8-A15 will not contain addresses until the data direction register for port 4 has been written with "1s" in the appropriate bits. These address lines will assert "1s" until made outputs by writing the data direction register.

TABLE 4 — INTERNAL REGISTER AREA

Register	Address: Other Modes	Address: Mode 3
Port 1 Data Direction Register* * *	0000	D000
Port 2 Data Direction Register* * *	0001	D001
Port 1 Data Register	0002	D002
Port 2 Data Register	0003	D003
Port 3 Data Direction Register* * *	0004*	D004*
Port 4 Data Direction Register* * *	0005* *	D005* *
Port 3 Data Register	0006*	D006*
Port 4 Data Register	0007* *	D007* *
Timer Control and Status Register	0008	D008
Counter (High Byte)	0009	D009
Counter (Low Byte)	000A	D00A
Output Compare Register (High Byte)	000B	D00B
Output Compare Register (Low Byte)	000C	D00C
Input Capture Register (High Byte)	000D	D00D
Input Capture Register (Low Byte)	000E	D00E
Port 3 Control and Status Register	000F*	D00F*
Rate and Mode Control Register	0010	D010
Transmit/Receive Control and Status Register	0011	D011
Receive Data Register	0012	D012
Transmit Data Register	0013	D013
RAM Control Register	0014	D014
Counter Alternate Address (High Byte)	0015	D015
Counter Alternate Address (Low Byte)	0016	D016
Timer Control Register 1	0017	D017
Timer Control Register 2	0018	D018
Timer Status Register	0019	D019
Output Compare Register 2 (High Byte)	001A	D01A
Output Compare Register 2 (Low Byte)	001B	D01B
Output Compare Register 3 (High Byte)	001C	D01C
Output Compare Register 3 (Low Byte)	001D	D01D
Input Capture Register 2 (High Byte)	001E	D01E
Input Capture Register 2 (Low Byte)	001F	D01F

*External addresses in modes 0, 1, 2, 3, 5, and 6 cannot be accessed in mode 5 (no IOS).

**External Addresses in Modes 0, 2, and 3.

***1 = Output, 0 = Input

EF6801U4/EF6803U4 INTERRUPTS

The EF6801 Family supports two types of interrupt requests: maskable and non-maskable. A non-maskable interrupt ($\overline{NMI}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the condition code register I bit and by individual enable bits. The I bit controls all maskable interrupts. Of the maskable interrupts, there are two types: $\overline{IRQ1}$ and $\overline{IRQ2}$. The programmable timer and serial communications interface use an internal $\overline{IRQ2}$ interrupt line, as shown in the block diagram. External devices and IS3 use $\overline{IRQ1}$. An $\overline{IRQ1}$ interrupt is serviced before $\overline{IRQ2}$ if both are pending.

NOTE

After reset, an $\overline{NMI}$ will not be serviced until the first program load of the stack pointer. Any $\overline{NMI}$ generated before this load will be remembered by the processor and serviced subsequent to the stack pointer load.

All $\overline{IRQ2}$ interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order and each is vectored to a separate location. All interrupt vector locations are shown in Table 5. In mode 0, reset and interrupt vectors are defined as $BFF0-$BFFF.

The interrupt flowchart is depicted in Figure 17 and is common to every interrupt excluding reset. During interrupt servicing, the program counter, index register, A accumulator, B accumulator, and condition code register are pushed to the stack. The I bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the program counter and instruction execution is resumed. Interrupt and $\overline{RESET}$ timing are illustrated in Figures 18 and 19.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

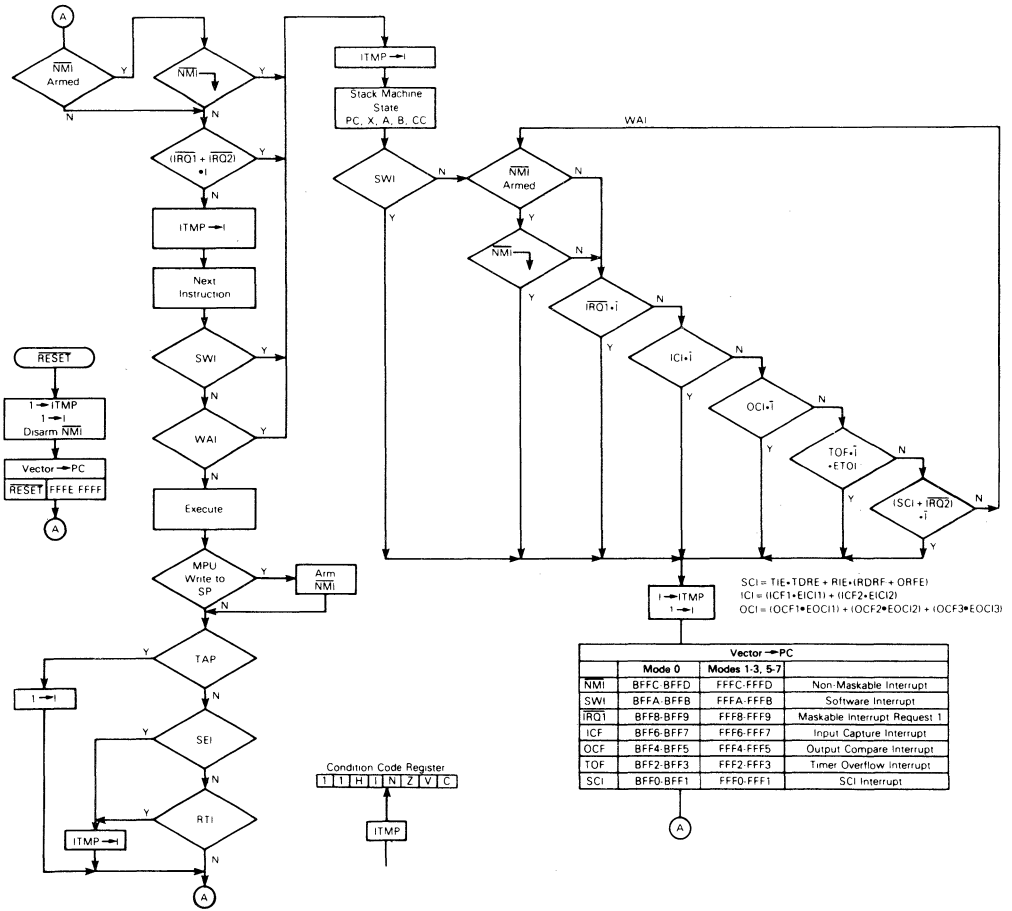
Mode 0 MSB	Mode 0 LSB	Modes 1-3, 5-7 MSB	Modes 1-3, 5-7 LSB	Interrupt* * *
BFFE	BFFF	FFFE	FFFF	$\overline{RESET}$
BFFC	BFFD	FFFC	FFFD	Non-Maskable Interrupt* *
BFFA	BFFB	FFFA	FFFB	Software Interrupt
BFF8	BFF9	FFF8	FFF9	Maskable Interrupt Request 1
BFF6	BFF7	FFF6	FFF7	Input Capture Flag*
BFF4	BFF5	FFF4	FFF5	Output Compare Flag*
BFF2	BFF3	FFF2	FFF3	Timer Overflow Flag*
BFF0	BFF1	FFF0	FFF1	Serial Communications Interface*

*$\overline{IRQ2}$ interrupt

**$\overline{NMI}$ must be armed (by accessng stack pointer) before an $\overline{NMI}$ is executed.

*** Mode 4 interrupt vectors are undefined.

FIGURE 17 — INTERRUPT FLOWCHART



	Vector → PC		
	Mode 0	Modes 1-3, 5-7	
NMI	BFFC-BFFD	FFFC-FFFD	Non-Maskable Interrupt
SWI	BFFA-BFFB	FFFA-FFFB	Software Interrupt
IRQ1	BFF8-BFF9	FFF8-FFF9	Maskable Interrupt Request 1
ICF	BFF6-BFF7	FFF6-FFF7	Input Capture Interrupt
OCF	BFF4-BFF5	FFF4-FFF5	Output Compare Interrupt
TOF	BFF2-BFF3	FFF2-FFF3	Timer Overflow Interrupt
SCI	BFF0-BFF1	FFF0-FFF1	SCI Interrupt

FIGURE 18 – INTERRUPT SEQUENCE

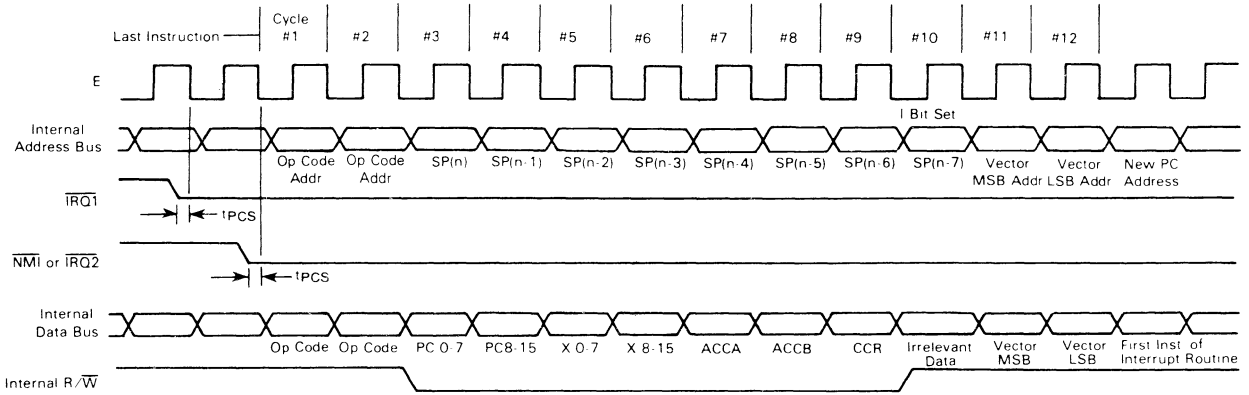
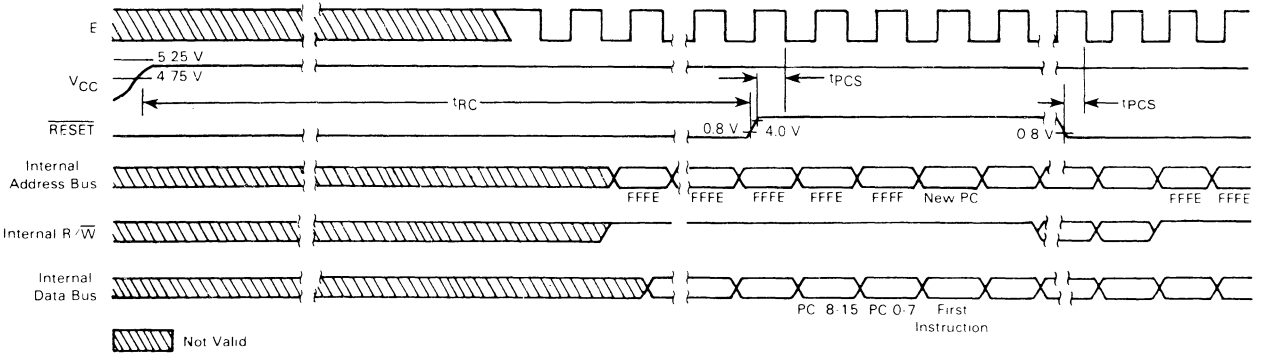


FIGURE 19 – RESET TIMING



FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power to a large portion of the MCU. The power supply should provide +5 volts (±5%) to V_{CC} and V_{SS} should be tied to ground. Total power dissipation (including V_{CC} standby) will not exceed P_D milliwatts.

V_{CC} STANDBY

V_{CC} standby provides power to the standby portion ($40 through $5F in all modes except mode 3 which is $D040 through $D05F) of the RAM and the STBY PWR and RAME bits of the RAM control register. Voltage requirements depend on whether the device is in a power-up or power-down state. In the power-up state, the power supply should provide +5 volts (±5%) and must reach V_{SB} volts before $\overline{RESET}$ reaches 4.0 volts. During power down, V_{CC} standby must remain above V_{SBB} (minimum) to sustain the standby RAM and STBY PWR bit. While in power-down operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during power-down operation.

XTAL1 AND EXTAL 2

These two input pins interface either a crystal or TTL-compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58 MHz or 4.4336 MHz color burst TV crystals. A 20 pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL-compatible clock at 4 f_o with a duty cycle of 50% (±5%) with XTAL1 connected ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for f_{XTAL}. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in Figure 20.

$\overline{RESET}$

This input is used to reset the internal state of the device and provide an orderly startup procedure. During power up, $\overline{RESET}$ must be held below 0.8 volt : (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} standby reaches 4.75 volts. $\overline{RESET}$ must be held low at least three E cycles if asserted during power-up operation.

E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the device input clock frequency. It will drive one Schottky TTL load and 90 pF, and all data given in cycles is referenced to this clock unless otherwise noted.

$\overline{NMI}$ (NON-MASKABLE INTERRUPT)

An $\overline{NMI}$ negative edge requests an MCU interrupt sequence, but the current instruction will be completed before it responds to the request. The MCU will then begin an interrupt sequence. Finally, a vector is fetched from $FFFC and $FFFD ($BFFC and $BFFD in mode 0), transferred to the program counter, and instruction execution is resumed. $\overline{NMI}$ typically requires a 3.3 kΩ (nominal) resistor to V_{CC}. There is no internal $\overline{NMI}$ pullup resistor. $\overline{NMI}$ must be held low for at least one E cycle to be recognized under all conditions.

NOTE

After reset, an $\overline{NMI}$ will not be serviced until the first program load of the stack pointer. Any $\overline{NMI}$ generated before this load will remain pending by the processor.

$\overline{IRQ1}$ (MASKABLE INTERRUPT REQUEST 1)

$\overline{IRQ1}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I bit) in the condition code register is clear, the MCU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9 ($BFF8 and $BFF9 in mode 0), transferred to the program counter, and instruction execution is resumed.

$\overline{IRQ1}$ typically requires an external 3.3 kΩ (nominal) resistor to V_{CC} for wire-OR applications. $\overline{IRQ1}$ has no internal pullup resistor.

SC1 AND SC2 (STROBE CONTROL 1 AND 2)

The function of SC1 and SC2 depends on the operating mode. SC1 is configured as an output in all modes except single-chip mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90 pF.

SC1 AND SC2 IN SINGLE-CHIP MODE – In single-chip mode, SC1 and SC2 are configured as an input and output, respectively, and both function as port 3 control lines. SC1 functions as IS3 and can be used to indicate that port 3 input data is ready or output data has been accepted. Three options associated with IS3 are controlled by the port 3 control and status register and are discussed in the port 3 description; refer to **P30-P37 (PORT 3).** If unused, IS3 can remain unconnected.

SC2 is configured as OS3 and can be used to strobe output data or acknowledge input data. It is controlled by output strobe select (OSS) in the port 3 control and status register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the port 3 data register. OS3 timing is shown in Figure 3.

SC1 AND SC2 IN EXPANDED NON-MULTIPLEXED MODE – In the expanded non-multiplexed mode, both SC1 and SC2 are configured as outputs. SC1 functions as input/output select (IOS) and is asserted only when $0100 through $01FF is sensed on the internal address bus.

SC2 is configured as read/write and is used to control the direction of data bus transfers. An MPU read is enabled when read/write and E are high.

SC1 AND SC2 IN EXPANDED MULTIPLEXED MODE – In the expanded multiplexed modes, both SC1 and SC2 are configured as outputs. SC1 functions as address strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by address strobe captures the lower address on the negative edge, as shown in Figure 13.

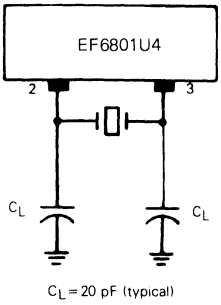
FIGURE 20 – EF6801U4/EF6803U4 FAMILY OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

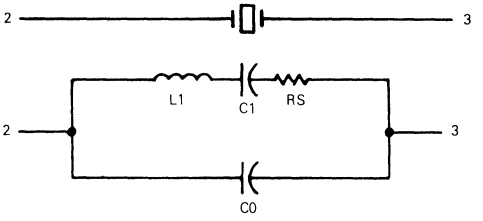
Nominal Crystal Parameters*

	3.58 MHz	4.00 MHz	5.0 MHz
RS	60 Ω	50 Ω	30-50 Ω
C0	3.5 pF	6.5 pF	4-6 pF
C1	0.015 pF	0.025 pF	0.01-0.02 pF
Q	>40 K	>30 K	>20 K

*NOTE: These are representative AT-cut crystal parameters only. Crystals of other types of cut may also be used

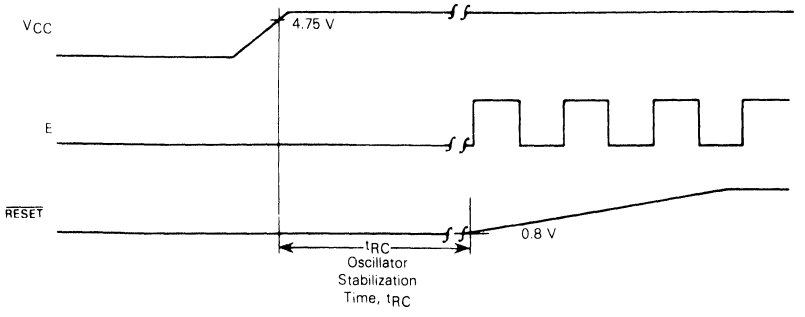


C_L = 20 pF (typical)



Equivalent Circuit

(b) Oscillator Stabilization Time (t_{RC})



SC2 is configured as read/write and is used to control the direction of data bus transfers. An MPU read is enabled when read/write and E are high.

P10-P17 (PORT 1)

Port 1 is a mode independent 8-bit I/O and timer port. Each line can be configured as either an input or output as defined by the port 1 data direction register. Port 1 bits 0, 1, and 2 (P10, P11, and P12) can also be used to exercise one input edge function and two output compare functions of the timer. The TTL compatible three-state buffers can drive one Schottky TTL load and 30 pF, Darlington transistors, or CMOS devices using external pullup resistors. It is configured as a data input port during $\overline{\text{RESET}}$. Unused pins can remain unconnected.

P20-P24 (PORT 2)

Port 2 is a mode-independent, 5-bit, multipurpose I/O port. The voltage levels present on P20, P21, and P22 on the rising edge of $\overline{\text{RESET}}$ determine the operating mode of the MCU. The entire port is then configured as a data input port. The port 2 lines can be selectively configured as data output lines by setting the appropriate bits in the port 2 data direction register. The port 2 data register is used to move data through the port. However, if P21 is configured as an output, it is tied to the timer output compare 1 function and cannot be used to provide output from the port 2 data register unless output enable 1 (OE1) is cleared in timer control register 1.

Port 2 can also be used to provide an interface for the serial communications interface and the timer input edge function. These configurations are described in **SERIAL COMMUNICATIONS INTERFACE** and **PROGRAMMABLE TIMER.**

The port 2 three-state TTL-compatible output buffers are capable of driving one Schottky TTL load and 30 pF, or CMOS devices using external pullup resistors.

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

P30-P37 (PORT 3)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address/data bus depending on the operating mode. The TTL compatible three-state output buffers can drive one Schottky TTL load and 90 pF. Unused lines can remain unconnected.

PORT 3 IN SINGLE-CHIP MODE – Port 3 is an 8-bit I/O port in the single-chip mode with each line configured by the port 3 data direction register. There are also two lines, IS3 and OS3, which can be used to control port 3 data transfers.

Three port 3 options are controlled by the port 3 control and status register and are available only in single-chip mode: 1) port 3 input data can be latched using IS3 as a control signal, 2) OS3 can be generated by either an MPU read or write to the port 3 data register, and 3) an $\overline{\text{IRQ1}}$ interrupt can be enabled by an IS3 negative edge. Port 3 latch timing is shown in Figure 4.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3 Flag	IS3 $\overline{\text{IRQ1}}$	X	OSS	Latch Enable	X	X	X	$0F

Bits 0-2 Not used.

Bit 3 **Latch Enable** – This bit controls the input latch for port 3. If set, input data is latched by an IS3 negative edge. The latch is transparent after a read of the port 3 data register. Latch enable is cleared during reset.

Bit 4 **OSS (Output Strobe Select)** – This bit determines whether OS3 will be generated by a read or write of the port 3 data register. When clear, the strobe is generated by a read; when set, it is generated by a write. OSS is cleared during reset.

Bit 5 Not used.

Bit 6 **IS3 $\overline{\text{IRQ1}}$ Enable** – When set, an $\overline{\text{IRQ1}}$ interrupt will be enabled whenever the IS3 flag is set; when clear, the interrupt is inhibited. This bit is cleared during reset.

Bit 7 **IS3 Flag** – This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the port 3 data register or during reset.

PORT 3 IN EXPANDED NON-MULTIPLEXED MODE – Port 3 is configured as a bidirectional data bus (D7-D0) in the expanded non-multiplexed mode. The direction of data transfers is controlled by read/write (SC2). Data is clocked by E (enable).

PORT 3 IN EXPANDED MULTIPLEXED MODE – Port 3 is configured as a time multiplexed address (A7-A0) and data bus (D7-D0) in the expanded multiplexed mode where address strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent bus conflicts.

P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF, and is the only port with internal pullup resistors. Unused lines can remain unconnected.

PORT 4 IN SINGLE-CHIP MODE – In single-chip mode, port 4 functions as an 8-bit I/O port with each line configured by the port 4 data direction register. Internal pullup resistors allow the port to directly interface with CMOS at 5-volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

PORT 4 IN EXPANDED NON-MULTIPLEXED MODE – Port 4 is configured from reset as an 8-bit input port where the port 4 data direction register can be written to provide any or all of eight address lines A0 to A7. Internal pullup resistors pull the lines high until the port 4 data direction register is configured.

PORT 4 IN EXPANDED MULTIPLEXED MODE – In all expanded multiplexed modes except modes 1 and 6, port 4 functions as half of the address bus and provides A8 to A15. In modes 1 and 6, the port is configured from reset as an 8-bit parallel input port where the port 4 data direction register can be written to provide any or all of upper address lines A8 to A15. Internal pullup resistors pull the lines high until the port 4 data direction register is configured where bit 0 controls A8.

RESIDENT MEMORY

The EF6801U4 provides 4096 bytes of on-chip ROM and 192 bytes of on-chip RAM.

Thirty-two bytes of the RAM are powered through the V_{CC} standby pin and are maintainable during V_{CC} power down. This standby portion of the RAM consists of 32 bytes located from $40 through $5F in all modes except mode 3 which is $D040 through $D05F.

Power must be supplied to V_{CC} standby if the internal RAM is to be used regardless of whether standby power operation is anticipated.

The RAM is controlled by the RAM control register.

RAM CONTROL REGISTER ($14)

The RAM control register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during power-down operation. It is intended that RAME be cleared and STBY PWR be set as part of a power-down procedure.

RAM CONTROL REGISTER

7	6	5	4	3	2	1	0	
STBY PWR	RAME	X	X	X	X	X	X	$14

Bits 0-5 Not used.

Bit 6 **RAM Enable** – This read/write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of $\overline{RESET}$. If RAME is clear, any access to a RAM address is external. If RAME is set, the RAM is included in the internal map.

Bit 7 **Standby Power** – This bit is a read/write status bit which when cleared indicates that V_{CC} standby has decreased sufficiently below V_{SBB} (minimum) to make data in the standby RAM suspect. It can be set only by software and is not affected during reset.

PROGRAMMABLE TIMER

The programmable timer can be used to perform measurements on two separate input waveforms while independently generating three output waveforms. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 21.

COUNTER ($09:0A), ($15, $16)

The key timer element is a 16-bit free-running counter which is incremented by E (enable). It is cleared during reset and is read-only with one exception: in mode 0 a write to the counter ($09) will configure it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. The TOF is set whenever the counter contains all ones. If ETOI is set, an interrupt will occur when the TOF is set. The counter may also be read as $15 and $16 to avoid inadvertently clearing the TOF.

OUTPUT COMPARE REGISTERS ($0B:0C), ($1A:1B), ($1C:1D)

The three output compare registers are 16-bit read/write registers, each used to control an output waveform or provide an arbitrary time-out flag. They are compared with the free-running counter during the negative half of each E cycle. When a match occurs, the corresponding output compare flag (OCF) is set and the corresponding output level (OLVL) is clocked to an output level register. If both the corresponding output enable bit and data direction register bit are set, the value represented in the output level register will appear on the corresponding port pin. The appropriate OLVL bit can then be changed for the next compare.

The function is inhibited for one cycle after a write to its high byte ($0B, $1A, or $1C) to ensure a valid compare after a double byte write. Writes can be made to either byte of the output compare register without affecting the other byte. The OLVL value will be clocked out independently of whether the OCF had previously been cleared. The output compare registers are set to $FFFF during reset.

INPUT CAPTURE REGISTERS ($0D:0E), ($1E:1F)

The two input capture registers are 16-bit read-only registers used to store the free-running counter when a "proper" input transition occurs as defined by the corresponding input edge bit (IEDG1 or IEDG2). The input pin's data direction register should be configured as an input, but the edge detect circuit always senses P10 and P20 even when configured as an output. The counter value will be latched into the input capture registers on the second negative edge of the E clock following the transition.

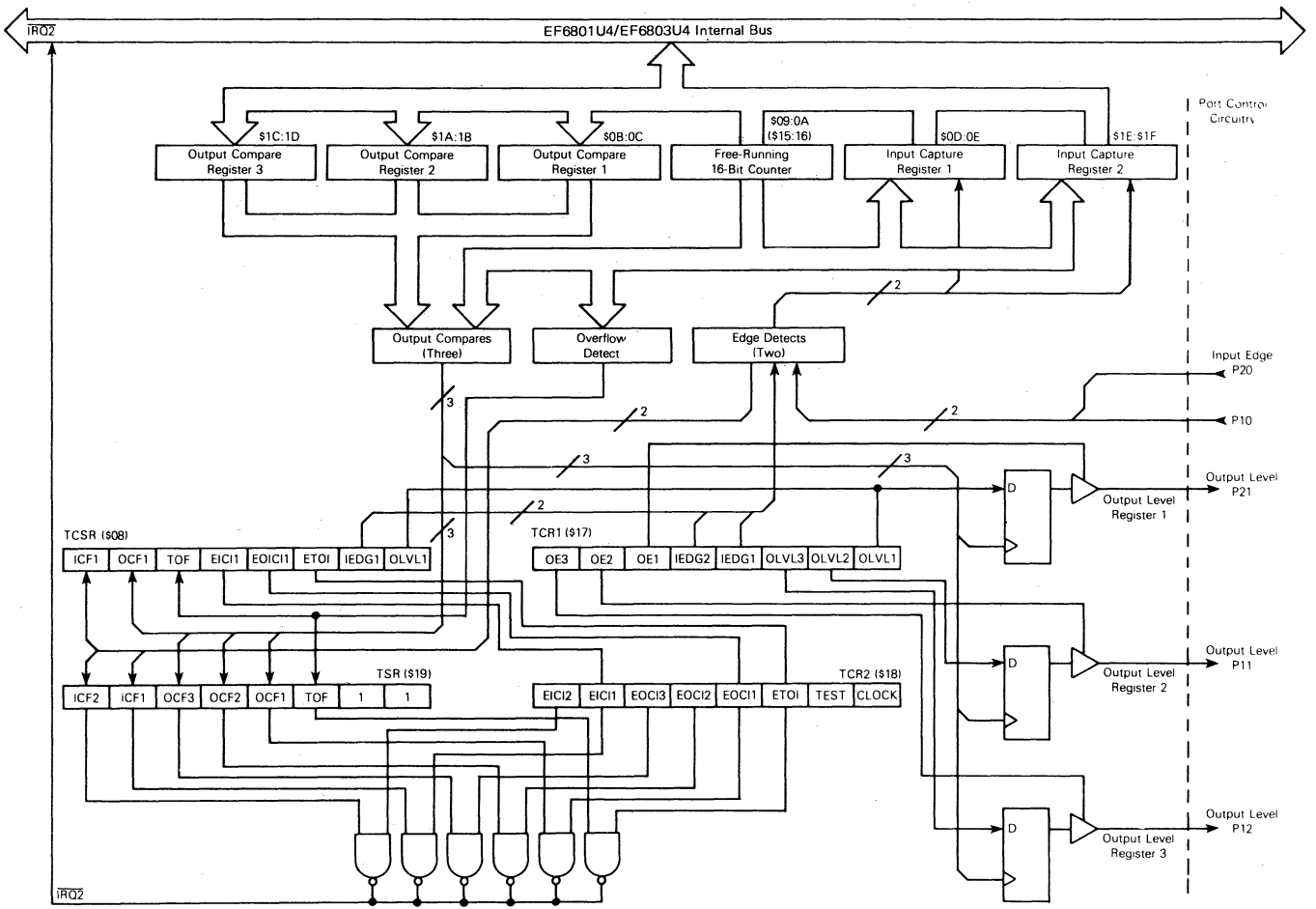
An input capture can occur independently of ICF; the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTERS

Four registers are used to provide the EF6801U4/EF6803U4 with control and status information about the three output compare functions, the timer overflow function, and the two input edge functions of the timer. They are:

Timer Control and Status Register (TCSR)
Timer Control Register 1 (TCR1)
Timer Control Register 2 (TCR2)
Timer Status Register (TSR)

FIGURE 21 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER



TIMER CONTROL AND STATUS REGISTER (TCSR) ($08) – The timer control and status register is an 8-bit register of which all bits are readable, while only bits 0-4 can be written. All the bits in this register are also accessible through the two timer control registers and the timer status register. The three most significant bits provide the timer status and indicate if:

1. a proper level transition has been detected at P20,
2. a match has occurred between the free-running counter and output compare register 1, or
3. the free-running counter has overflowed.

Each of the three events can generate an $\overline{IRQ2}$ interrupt and is controlled by an individual enable bit in the TCSR.

TIMER CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
ICF1	OCF1	TOF	EICI1	EOCI1	ETOI	IEDG1	OLVL1	$08

Bit 0 **Output Level 1** – OLVL1 is clocked to output level register 1 by a successful output compare and will appear at P21 if bit 1 of the port 2 data direction register is set and the OE1 control bit in timer control register 1 is set. OLVL1 and output level register 1 are cleared during reset. Refer to **TIMER CONTROL REGISTER 1 (TCR1) ($17)**.

Bit 1 **Input Edge 1** – IEDG1 is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register 1:
IEDG1 = 0 transfer on a negative-edge
IEDG1 = 1 transfer on a positive-edge
Refer to **TIMER CONTROL REGISTER 1 (TCR1) ($17)**.

Bit 2 **Enable Timer Overflow Interrupt** – When set, an $\overline{IRQ2}$ interrupt will be generated when the timer overflow flag is set; when clear, the interrupt is inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) ($18)**.

Bit 3 **Enable Output Compare Interrupt 1** – When set, an $\overline{IRQ2}$ interrupt will be generated when output compare flag 1 is set; when clear, the interrupt is inhibited. EOCI1 is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) ($18)**.

Bit 4 **Enable Input Capture Interrupt 1** – When set, an $\overline{IRQ2}$ interrupt will be generated when input capture flag 1 is set; when clear, the interrupt is inhibited. EICI1 is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) ($18).**

Bit 5 **Timer Overflow Flag** – The TOF is set when the counter contains all ones ($FFFF). It is cleared by reading the TCSR or the TSR (with TOF set) and the counter high byte ($09), or during reset. Refer to **TIMER STATUS REGISTER (TSR) ($19)**.

Bit 6 **Output Compare Flag 1** – OCF1 is set when output compare register 1 matches the free-running counter. OCF1 is cleared by reading the TCSR or the TSR (with OCF1 set) and then writing to output compare register 1 ($0B or $0C), or during reset. Refer to **TIMER STATUS REGISTER (TSR) ($19)**.

Bit 7 **Input Capture Flag** – ICF1 is set to indicate that a proper level transition has occurred; it is cleared by reading the TCSR or the TSR (with ICF1 set) and the input capture register 1 high byte ($0D), or during reset. Refer to **TIMER STATUS REGISTER (TSR) ($19)**.

TIMER CONTROL REGISTER 1 (TCR1) ($17) – Timer control register 1 is an 8-bit read/write register which contains the control bits for interfacing the output compare and input capture registers to the corresponding I/O pins.

TIMER CONTROL REGISTER 1

7	6	5	4	3	2	1	0	
OE3	OE2	OE1	IEDG2	IEDG1	OLVL3	OLVL2	OLVL1	$17

Bit 0 **Output Level 1** – OLVL1 is clocked to output level register 1 by a successful output compare and will appear at P21 if bit 1 of the port 2 data direction register is set and the OE1 control bit is set. OLVL1 and output level register 1 are cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 1 **Output Level 2** – OLVL2 is clocked to output level register 2 by a successful output compare and will appear at P11 if bit 1 of port 1 data direction register is set and the OE2 control bit is set. OLVL2 and output level register 2 are cleared during reset.

Bit 2 **Output Level 3** – OLVL3 is clocked to output level register 3 by a successful output compare and will appear at P12 if bit 2 of port 1 data direction register is set and the OE3 control bit is set. OLVL3 and output level register 3 are cleared during reset.

Bit 3 **Input Edge 1** – IEDG1 is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register 1.
IEDG1 = 0 transfer on a negative-edge
IEDG1 = 1 transfer on a positive-edge
Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 4 **Input Edge 2** – IEDG2 is cleared during reset and controls which level transition on P10 will trigger a counter transfer to input capture register 2.
IEDG2 = 0 transfer on a negative-edge
IEDG2 = 1 transfer on a positive-edge

Bit 5 **Output Enable 1** – OE1 is set during reset and enables the contents of output level register 1 to be connected to P21 when bit 1 of port 2 data direction register is set.
OE1 = 0 port 2 bit 1 data register output
OE1 = 1 output level register 1

Bit 6 **Output Enable 2** – OE2 is cleared during reset and enables the contents of output level register 2 to be connected to P11 when bit 1 of port 1 data direction register is set.
OE2 = 0 port 1 bit 1 data register output
OE2 = 1 output level register 2

Bit 7 **Output Enable 3** – OE3 is cleared during reset and enables the contents of output level register 3 to be connected to P12 when bit 2 of port 1 data direction register is set

OE3 = 0 port 1 bit 2 data register output

OE3 = 1 output level register 3

TIMER CONTROL REGISTER 2 (TCR2) ($18) – Timer control register 2 is an 8-bit read/write register (except bits 0 and 1) which enable the interrupts associated with the free-running counter, the output compare registers, and the input capture registers. In test mode 0, two more bits (clock and test) are available for checking the timer.

TIMER CONTROL REGISTER 2
(Non-Test Modes)

7	6	5	4	3	2	1	0	
EICI2	EICI1	EOCI3	EOCI2	EOCI1	ETOI	1	1	$18

Bits 0-1 **Read-Only Bits** – When read, these bits return a value of 1. Refer to **TIMER CONTROL REGISTER 2 (Test Mode)**.

Bit 2 **Enable Timer Overflow Interrupt** – When set, an $\overline{IRQ2}$ interrupt will be generated when the timer overflow flag is set; when clear, the interrupt is inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 3 **Enable Output Compare Interrupt 1** – When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 1 is set; when clear, the interrupt is inhibited. EOCI1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 4 **Enable Output Compare Interrupt 2** – When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 2 is set; when clear, the interrupt is inhibited. EOCI2 is cleared during reset.

Bit 5 **Enable Output Compare Interrupt 3** – When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 3 is set; when clear, the interrupt is inhibited. EOCI3 is cleared during reset.

Bit 6 **Enable Input Capture Interrupt 1** – When set, an $\overline{IRQ2}$ interrupt will be generated when the input capture flag 1 is set; when clear, the interrupt is inhibited. EICI1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 7 **Enable Input Capture Interrupt 2** – When set, an $\overline{IRQ2}$ interrupt will be generated when the input capture flag 2 is set; when clear, the interrupt is inhibited. EICI2 is cleared during reset.

The timer test bits (test and clock) allow the free-running counter to be tested as two separate 8-bit counters to speed testing.

TIMER CONTROL REGISTER 2
(Test Mode)

7	6	5	4	3	2	1	0	
EICI2	EICI1	EOCI3	EOCI2	EOCI1	ETOI	TEST	CLOCK	$18

Bit 0 **CLOCK** – The CLOCK control bit selects which half of the 16-bit free-running counter (MSB or LSB) should be clocked with E. The CLOCK bit is a read/write bit only in mode 0 and is set during reset.

CLOCK = 0 – Only the eight most significant bits of the free-running counter run with TEST = 0.

CLOCK = 1 – Only the eight least significant bits of the free-running counter run when TEST = 0.

Bit 1 **TEST** – the TEST control bit enables the timer test mode. TEST is a read/write bit in mode 0 and is set during reset.

TEST = 0 – Timer test mode enabled:

a) The timer LSB latch is transparent which allows the LSB to be read independently of the MSB.

b) Either the MSB or the LSB of the timer is clocked by E, as defined by the CLOCK bit.

TEST = 1 – Timer test mode disabled.

Bits 2-7 See **TIMER CONTROL REGISTER 2 (Non-Test Modes)**. (These bits function the same as in the non-test modes.)

TIMER STATUS REGISTER (TSR) ($19) – The timer status register is an 8-bit read-only register which contains the flags associated with the free-running counter, the output compare registers, and the input capture registers.

TIMER STATUS REGISTER

7	6	5	4	3	2	1	0	
ICF2	ICF1	OCF3	OCF2	OCF1	TOF	1	1	$19

Bits 0-1 Not used.

Bit 2 **Timer Overflow Flag** – The TOF is set when the counter contains all ones ($FFFF). It is cleared by reading the TSR or the TCSR (with TOF set) and then the counter high byte ($09), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 3 **Output Compare Flag 1** – OCF1 is set when output compare register 1 matches the free-running counter. OCF1 is cleared by reading the TSR or the TCSR (with OCF1 set) and then writing to output compare register 1 ($0B or $0C), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 4 **Output Compare Flag 2** – OCF2 is set when output compare register 2 matches the free-running counter. OCF2 is cleared by reading the TSR (with OCF2 set) and then writing to output compare register 2 ($1A or $1B), or during reset.

Bit 5 **Output Compare Flag 3** – OCF3 is set when output compare register 3 matches the free-running counter. OCF3 is cleared by reading the TSR (with OCF3 set) and then writing to output compare register 3 ($1C or $1D), or during reset.

Bit 6 **Input Capture Flag 1** – ICF1 is set to indicate that a proper level transition has occurred; it is cleared by reading the TSR or the TCSR (with ICF1 set) and the input capture register 1 high byte ($0D), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 7 **Input Capture Flag 2** – ICF2 is set to indicate that a proper level transition has occurred; it is cleared by reading the TSR (with ICF2 set) and the input capture register 2 high byte ($1E), or during reset.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous serial communications interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and bi-phase and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

In a typical serial loop multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive ones or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

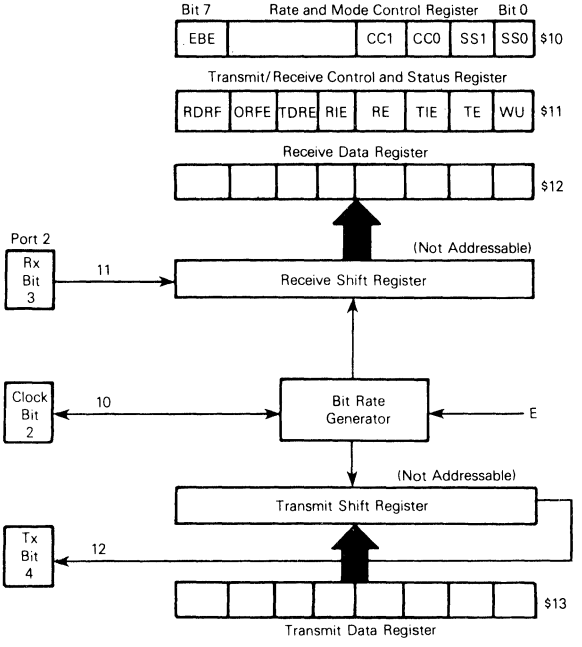
The following features of the SCI are programmable:

- Format: standard mark/space (NRZ) or bi-phase
- Clock: external or internal bit rate clock
- Baud: one of eight per E clock frequency or external clock (×8 desired baud)
- Wake-Up Feature: enabled or disabled
- Interrupt Requests: enabled individually for transmitter and receiver
- Clock Output: internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The serial communications interface includes four addressable registers as depicted in Figure 22. It is controlled by the rate and mode control register and the transmit/receive control and status register. Data is transmitted and received utilizing a write-only transmit register and a read-only receive register. The shift registers are not accessible to software.

FIGURE 22 – SCI REGISTERS



RATE AND MODE CONTROL REGISTER (RMCR) ($10)

– The rate and mode control register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of five write-only bits which are cleared during reset. The two least significant bits in conjunction with bit 7 control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER

7	6	5	4	3	2	1	0	
EBE	X	X	X	CC1	CC0	SS1	SS0	$10

Bit 1:Bit 0 **SS1:SS0 Speed Select** – These two bits select the baud when using the internal clock. Eight rates may be selected (in conjunction with bit 7) which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

Bit 3:Bit 2 **CC1:CC0 Clock Control and Format Select** – These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

Bits 4-6 Not used.

Bit 7 **EBE Enhanced Baud Enable** – EBE selects the standard EF6801 baud rates when clear and the additional baud rates when set (Table 6). This bit is cleared by reset and is a write-only control bit.

EBE = 0 standard EF6801 baud rates
EBE = 1 additional baud rates

If both CC1 and CC0 are set, an external TTL-compatible clock must be connected to P22 at eight times (8×) the desired bit rate, but not greater than E, with a duty cycle of 50% (±10%). If CC1:CC0 = 10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE

The source of SCI internal bit rate clock is the timer free-running counter. An MPU write to the counter in mode 0 can disturb serial operations.

TABLE 6 – SCI BIT TIMES AND RATES

EBE	SS1:SS0		4 f_o → E	2.4576 MHz 614.4 kHz		4.0 MHz 1.0 MHz		4.9152 MHz 1.2288 MHz	
				Baud	Time	Baud	Time	Baud	Time
0	0	0	÷16	38400.0	26 μs	62500.0	16.0 μs	76800.0	13.0 μs
0	0	1	÷128	4800.0	208.3 μs	7812.5	128.0 μs	9600.0	104.2 μs
0	1	0	÷1024	600.0	1.67 ms	976.6	1.024 ms	1200.0	833.3 μs
0	1	1	÷4096	150.0	6.67 ms	244.1	4.096 ms	300.0	3.33 ms
1	0	0	÷64	9600.0	104.2 μs	15625.0	64 μs	19200.0	52.0 μs
1	0	1	÷256	2400.0	416.6 μs	3906.3	256 μs	4800.0	208.3 μs
1	1	0	÷512	1200.0	833.3 μs	1953.1	512 μs	2400.0	416.6 μs
1	1	1	÷2048	300.0	3.33 ms	488.3	2.05 ms	600.0	1.67 ms
External (P22)*				76800.0	13.0 μs	125000.0	8.0 μs	153600.0	6.5 μs

*Using maximum clock rate

TABLE 7 – SCI FORMAT AND CLOCK SOURCE CONTROL

CC1:CC0	Format	Clock Source	Port 2 Bit 2
00	Bi-Phase	Internal	Not Used
01	NRZ	Internal	Not Used
10	NRZ	Internal	Output
11	NRZ	External	Input

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR) ($11) The transmit/receive control and status register controls the transmitter, receiver, wake-up feature, and two individual interrupts, and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by $\overline{\text{RESET}}$

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$11

Bit 0 **"Wake-Up" on Idle Line** – When set, WU enables the wake-up function; it is cleared by ten consecutive ones or during reset. WU will not be set if the line is idle. Refer to **WAKE-UP FEATURE.**

Bit 1 **Transmit Enable** – When set, P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive ones is transmitted. TE is cleared during reset.

Bit 2 **Transmit Interrupt Enable** – When set, an $\overline{\text{IRQ2}}$ is set; when clear, the interrupt is inhibited. TE is cleared during reset.

Bit 3 **Receive Enable** – When set, the P23 DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared during reset.

Bit 4 **Receiver Interrupt Enable** – When set, an $\overline{\text{IRQ2}}$ interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared during reset.

Bit 5 **Transmit Data Register Empty** – TDRE is set when the transmit data register is transferred to the output serial shift register or during reset. It is cleared by reading the TRCSR (with TDRE set) and then writing to the transmit data register. Additional data will be transmitted only if TDRE has been cleared.

Bit 6 **Overrun Framing Error** – If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF: if RDRF is set, then an overrun has occurred; otherwise, a framing error has been detected. Data is not transferred to the receive data register in an overrun condition. Unframed data causing a framing error is transferred to the receive data register. However, subsequent data transfer is blocked until the framing error flag is cleared. ORFE is cleared by reading the TRCSR (with ORFE set) then the receive data register, or during reset.

Bit 7 **Receive Data Register Full** – RDRF is set when the input serial shift register is transferred to the receive data register, or during reset.

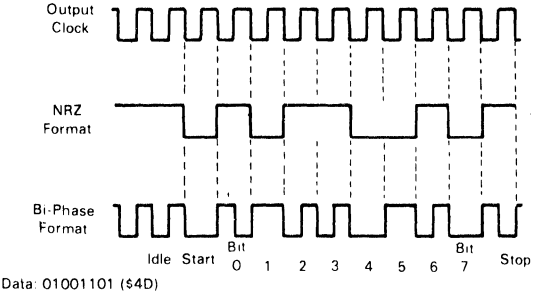
SERIAL OPERATIONS

The SCI is initialized by writing control bytes first to the rate and mode control register and then to the transmit/receive control and status register. When TE is set, the output of the transmit serial shift register is connected to P24 and serial output is initiated by transmitting a 9-bit preamble of ones.

At this point, one of two situations exists: 1) if the transmit data register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line; or 2) if a byte has been written to the transmit data register (TDRE = 0), it will be transferred to the output serial shift register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0), and a stop bit (1) will be transmitted. If TDRE is still set when the next byte transfer occurs, ones will be sent until more data is provided. In bi-phase format, the output toggles at the start of each bit and at half-bit time when a one is sent. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. SCI data formats are illustrated in Figure 23.

FIGURE 23 – SCI DATA FORMATS



Data: 01001101 ($4D)

INSTRUCTION SET

The EF6801U4/EF6803U4 is directly source compatible with the EF6801 and upward source and object code compatible with the EF6800. Execution times of key instructions have been reduced and several instructions have been added, including a hardware multiply. A list of new operations added to the EF6800 instruction set is shown in Table 1.

In addition, two special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the program counter to increment like a 16-bit counter causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in Table 8. There are 220 valid machine codes, 34* unassigned codes, and 2 codes reserved for test purposes.

TABLE 8 — CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
00	•				34	DES	INHER	3	1	68	ASL	INDXD	6	2	9C	CPX	DIR	5	2	D0	SUBB	DIR	3	2
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2	D1	CMPB		3	2
02	•				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	D2	SBCB		3	2
03	•				37	PSHB		3	1	6B	•				9F	STS	DIR	4	2	D3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	INDXD	4	2	D4	ANDB		3	2
05	ASLD		3	1	39	RTS		5	1	6D	TST		6	2	A1	CMPA		4	2	D5	BITB		3	2
06	TAP		2	1	3A	ABX		3	1	6E	JMP		3	2	A2	SBCA		4	2	D6	LDAB		3	2
07	TPA		2	1	3B	RTI		10	1	6F	CLR	INDXD	6	2	A3	SUBD		6	2	D7	STAB		3	2
08	INX		3	1	3C	PSHX		4	1	70	NEG	EXTND	6	3	A4	ANDA		4	2	D8	EORB		3	2
09	DEX		3	1	3D	MUL		10	1	71	•				A5	BITA		4	2	D9	ADCB		3	2
0A	CLV		2	1	3E	WAI		9	1	72	•				A6	LDAA		4	2	DA	ORAB		3	2
0B	SEV		2	1	3F	SWI		12	1	73	COM		6	3	A7	STAA		4	2	DB	ADDB		3	2
0C	CLC		2	1	40	NEGA		2	1	74	LSR		6	3	A8	EORA		4	2	DC	LDD		4	2
0D	SEC		2	1	41	•				75	•				A9	ADCA		4	2	DD	STD		4	2
0E	CLI		2	1	42	•				76	ROR		6	3	AA	ORAA		4	2	DE	LDX		4	2
0F	SEI		2	1	43	COMA		2	1	77	ASR		6	3	AB	ADDA		4	2	DF	STX	DIR	4	2
10	SBA		2	1	44	LSRA		2	1	78	ASL		6	3	AC	CPX		6	2	E0	SUBB	INDXD	4	2
11	CBA		2	1	45	•				79	ROL		6	3	AD	JSR		6	2	E1	CMPB		4	2
12	•				46	RORA		2	1	7A	DEC		6	3	AE	LDS		5	2	E2	SBCB		4	2
13	•				47	ASRA		2	1	7B	•				AF	STS	INDXD	5	2	E3	ADDD		6	2
14	•				48	ASLA		2	1	7C	INC		6	3	B0	SUBA	EXTND	4	3	E4	ANDB		4	2
15	•				49	ROLA		2	1	7D	TST		6	3	B1	CMPA		4	3	E5	BITB		4	2
16	TAB		2	1	4A	DECA		2	1	7E	JMP		3	3	B2	SBCA		4	3	E6	LDAB		4	2
17	TBA		2	1	4B	•				7F	CLR	EXTND	6	3	B3	SUBD		6	3	E7	STAB		4	2
18	•				4C	INCA		2	1	80	SUBA	IMMED	2	2	B4	ANDA		4	3	E8	EORB		4	2
19	DAA	INHER	2	1	4D	TSTA		2	1	81	CMPA		2	2	B5	BITA		4	3	E9	ADCB		4	2
1A	•				4E	T				82	SBCA		2	2	B6	LDAA		4	3	EA	ORAB		4	2
1B	ABA	INHER	2	1	4F	CLRA		2	1	83	SUBD		4	3	B7	STAA		4	3	EB	ADDB		4	2
1C	•				50	NEGB		2	1	84	ANDA		2	2	B8	EORA		4	3	EC	LDD		5	2
1D	•				51	•				85	BITA		2	2	B9	ADCA		4	3	ED	STD		5	2
1E	•				52	•				86	LDAA		2	2	BA	ORAA		4	3	EE	LDX		5	2
1F	•				53	COMB		2	1	87	•				BB	ADDA		4	3	EF	STX	INDXD	5	2
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	BC	CPX		6	3	F0	SUBB	EXTND	4	3
21	BRN		3	2	55	•				89	ADCA		2	2	BD	JSR		6	3	F1	CMPB		4	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	BE	LDS		5	3	F2	SBCB		4	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	BF	STS	EXTND	5	3	F3	ADDD		6	3
24	BCC		3	2	58	ASLB		2	1	8C	CPX	IMMED	4	3	C0	SUBB	IMMED	2	2	F4	ANDB		4	3
25	BCS		3	2	59	ROLB		2	1	8D	BSR	REL	6	2	C1	CMPB		2	2	F5	BITB		4	3
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		2	2	F6	LDAB		4	3
27	BEQ		3	2	5B	•				8F	•				C3	ADDD		4	3	F7	STAB		4	3
28	BVC		3	2	5C	INCB		2	1	90	SUBA	DIR	3	2	C4	ANDB		2	2	F8	EORB		4	3
29	BVS		3	2	5D	TSTB		2	1	91	CMPA		3	2	C5	BITB		2	2	F9	ADCB		4	3
2A	BPL		3	2	5E	T				92	SBCA		3	2	C6	LDAB		2	2	FA	ORAB		4	3
2B	BMI		3	2	5F	CLRB	INHER	2	1	93	SUBD		5	2	C7	•				FB	ADDB		4	3
2C	BGE		3	2	60	NEG	INDXD	6	2	94	ANDA		3	2	C8	EORB		2	2	FC	LDD		5	3
2D	BLT		3	2	61	•				95	BITA		3	2	C9	ADCB		2	2	FD	STD		5	3
2E	BGT		3	2	62	•				96	LDAA		3	2	CA	ORAB		2	2	FE	LDX		5	3
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2	FF	STX	EXTND	5	3
30	TSX	INHER	3	1	64	LSR		6	2	98	EORA		3	2	CC	LDD		3	3					
31	INS		3	1	65	•				99	ADCA		3	2	CD	•								
32	PULA		4	1	66	ROR		6	2	9A	ORAA		3	2	CE	LDX	IMMED	3	3					
33	PULB		4	1	67	ASR	INDXD	6	2	9B	ADDA		3	2	CF	•								

*UNDEFINED OP CODE

NOTES:

1. Addressing Modes
   INHER≡Inherent INDXD≡Indexed IMMED≡Immediate
   REL≡Relative EXTND≡Extended DIR≡Direct
2. Unassigned opcodes are indicated by "•" and should not be executed.
3. Codes marked by "T" force the PC to function as a 16-bit counter.

PROGRAMMING MODEL

A programming model for the EF6801U4/EF6803U4 is shown in Figure 8. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulators A and/or B. Other registers are defined as follows:

PROGRAM COUNTER – The program counter is a 16-bit register which always points to the next instruction.

STACK POINTER – The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random-access memory at a location defined by the programmer.

INDEX REGISTER – The index register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

ACCUMULATORS – The MPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

CONDITION CODE REGISTER – The condition code register indicates the results of an instruction and includes the following five condition bits: negative (N), zero (Z), overflow (V), carry/borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7, are read as ones.

ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of addressing modes for all instructions is presented in Tables 9, 10, 11, and 12 where execution times are provided in E cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 MHz, one E cycle is equivalent to one microsecond. A cycle-by-cycle description of bus activity for each instruction is provided in Table 14 and descriptions of selected instructions are shown in Figure 24.

IMMEDIATE ADDRESSING – The operand or "immediate byte(s)" is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

DIRECT ADDRESSING – The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, the 256-byte area is reserved for frequently referenced data.

EXTENDED ADDRESSING – The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

INDEXED ADDRESSING – The unsigned offset contained in the second byte of the instruction is added with carry to the index register and is used to reference memory without changing the index register. These are two byte instructions.

INHERENT ADDRESSING – The operand(s) is a register and no memory reference is required. These are single byte instructions.

RELATIVE ADDRESSING – Relative addressing is used only for branch instructions. If the branch condition is true, the program counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current program counter. This provides a branch range of −126 to +129 bytes from the first byte of the instruction. These are two byte instructions.

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 14 provides a detailed description of the information present on the address bus, data bus, and the read/write (R/$\overline{W}$) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Note that during MPU reads of internal locations, the resultant value will not appear on the external data bus except in mode 0. "High order" byte refers to the most significant byte of a 16-bit value.

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

Pointer Operations	MNEM	Immed			Direct			Index			Extnd			Inherent			Boolean/ Arithmetic Operation	Condition Codes 5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Compare Index Register	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X − M:M + 1	•	•	↕	↕	↕	↕
Decrement Index Register	DEX													09	3	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pointer	DES													34	3	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Register	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pointer	INS													31	3	1	1 SP + 1 → SP	•	•	•	•	•	•
Load Index Register	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	↕	↕	R	•
Load Stack Pointer	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	•	•	↕	↕	R	•
Store Index Register	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	↕	↕	R	•
Store Stack Pointer	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	↕	↕	R	•
Index Reg → Stack Pointer	TXS													35	3	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Register	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP − 1 → SP X_H → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 1 of 2)

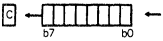
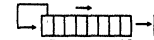
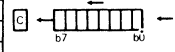
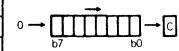
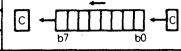
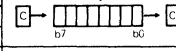
Accumulator and Memory Operations	MNEM	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes 5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A•M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B•M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕
Shift Left Double	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A•M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B•M	•	•	↕	↕	R	•
Compare Accumulators	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				M → M	•	•	↕	↕	R	S
	COMA													43	2	1	A → A	•	•	↕	↕	R	S
	COMB													53	2	1	B → B	•	•	↕	↕	R	S

TABLE 10 – ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 2 of 2)

Accumulator and Memory Operations		Immed			Direct			Index			Extend			Inher			Boolean	Condition Codes					
																		5	4	3	2	1	0
	MNEM	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Expression	H	I	N	Z	V	C
Decimal Adjust, A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M – 1 → M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A – 1 → A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B – 1 → B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Accumulators	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	•	•	↕	↕	R	•
Logical Shift, Left	LSL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	LSLA													48	2	1		•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	2		•	•	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R	↕	↕	↕
	LSRA													44	2	1		•	•	R	↕	↕	↕
	LSRB													54	2	1		•	•	R	↕	↕	↕
	LSRD													04	3	1		•	•	R	↕	↕	↕
Multiply	MUL													3D	10	1	A × B → D	•	•	•	•	•	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 – M → M	•	•	↕	↕	↕	↕
	NEGA													40	2	1	00 – A → A	•	•	↕	↕	↕	↕
	NEGB													50	2	1	00 – B → B	•	•	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	↕	↕
	ROLA													49	2	1		•	•	↕	↕	↕	↕
	ROLB													59	2	1		•	•	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	↕	↕
	RORA													46	2	1		•	•	↕	↕	↕	↕
	RORB													56	2	1		•	•	↕	↕	↕	↕
Subtract Accumulator	SBA													10	2	1	A – B → A	•	•	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A – M – C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B – M – C → B	•	•	↕	↕	↕	↕
Store Accumulators	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
	STD				DD	4	2	ED	5	2	FD	5	3				D → M: M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A – M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B – M → B	•	•	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D – M:M + 1 → D	•	•	↕	↕	↕	↕
Transfer Accumulator	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M – 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A – 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B – 00	•	•	↕	↕	R	R

The condition code register notes are listed after Table 12.

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	MNEM	Direct			Relative			Index			Extend			Inherent			Branch Test	Condition Code Reg. 5 H	4 I	3 N	2 Z	1 V	0 C
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#							
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•
Branch If Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR				8D	6	2											•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3				See Special Operations-Figure 24	•	•	•	•	•	•
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1	See Special Operations-Figure 24	•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

Operations	Inherent MNEM	Op	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	A → CCR	↕	↕	↕	↕	↕	↕
CCR → Accumulator A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

LEGEND

Op Operation Code (Hexadecimal)
~ Number of MPU Cycles
MSP Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
− Arithmetic Minus
• Boolean AND
X Arithmetic Multiply
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
M Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
↕ Affected
• Not Affected

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E-CYCLES

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
ABA	●	●	●	●	2	●
ABX	●	●	●	●	3	●
ADC	2	3	4	4	●	●
ADD	2	3	4	4	●	●
ADDD	4	5	6	6	●	●
AND	2	3	4	4	●	●
ASL	●	●	6	6	2	●
ASLD	●	●	●	●	3	●
ASR	●	●	6	6	2	●
BCC	●	●	●	●	●	3
BCS	●	●	●	●	●	3
BEQ	●	●	●	●	●	3
BGE	●	●	●	●	●	3
BGT	●	●	●	●	●	3
BHI	●	●	●	●	●	3
BHS	●	●	●	●	●	3
BIT	2	3	4	4	●	●
BLE	●	●	●	●	●	3
BLO	●	●	●	●	●	3
BLS	●	●	●	●	●	3
BLT	●	●	●	●	●	3
BMI	●	●	●	●	●	3
BNE	●	●	●	●	●	3
BPL	●	●	●	●	●	3
BRA	●	●	●	●	●	3
BRN	●	●	●	●	●	3
BSR	●	●	●	●	●	6
BVC	●	●	●	●	●	3
BVS	●	●	●	●	●	3
CBA	●	●	●	●	2	●
CLC	●	●	●	●	2	●
CLI	●	●	●	●	2	●
CLR	●	●	6	6	2	●
CLV	●	●	●	●	2	●
CMP	2	3	4	4	●	●
COM	●	●	6	6	2	●
CPX	4	5	6	6	●	●
DAA	●	●	●	●	2	●
DEC	●	●	6	6	2	●
DES	●	●	●	●	3	●
DEX	●	●	●	●	3	●
EOR	2	3	4	4	●	●
INC	●	●	6	6	●	●
INS	●	●	●	●	3	●

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
INX	●	●	●	●	3	●
JMP	●	●	3	3	●	●
JSR	●	5	6	6	●	●
LDA	2	3	4	4	●	●
LDD	3	4	5	5	●	●
LDS	3	4	5	5	●	●
LDX	3	4	5	5	●	●
LSL	●	●	6	6	2	●
LSLD	●	●	●	●	3	●
LSR	●	●	6	6	2	●
LSRD	●	●	●	●	3	●
MUL	●	●	●	●	10	●
NEG	●	●	6	6	2	●
NOP	●	●	●	●	2	●
ORA	2	3	4	4	●	●
PSH	●	●	●	●	3	●
PSHX	●	●	●	●	4	●
PUL	●	●	●	●	4	●
PULX	●	●	●	●	5	●
ROL	●	●	6	6	2	●
ROR	●	●	6	6	2	●
RTI	●	●	●	●	10	●
RTS	●	●	●	●	5	●
SBA	●	●	●	●	2	●
SBC	2	3	4	4	●	●
SEC	●	●	●	●	2	●
SEI	●	●	●	●	2	●
SEV	●	●	●	●	2	●
STA	●	3	4	4	●	●
STD	●	4	5	5	●	●
STS	●	4	5	5	●	●
STX	●	4	5	5	●	●
SUB	2	3	4	4	●	●
SUBD	4	5	6	6	●	●
SWI	●	●	●	●	12	●
TAB	●	●	●	●	2	●
TAP	●	●	●	●	2	●
TBA	●	●	●	●	2	●
TPA	●	●	●	●	2	●
TST	●	●	6	6	2	●
TSX	●	●	●	●	3	●
TXS	●	●	●	●	3	●
WAI	●	●	●	●	9	●

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 1 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Operand Data
LDS LDX LDD	3	1 2 3	Opcode Address Opcode Address + 1 Opcode Address + 2	1 1 1	Opcode Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Opcode Address + 2 Address Bus FFFF	1 1 1 1	Opcode Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
DIRECT					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Opcode Address Opcode Address + 1 Address of Operand	1 1 1	Opcode Address of Operand Operand Data
STA	3	1 2 3	Opcode Address Opcode Address + 1 Destination Address	1 1 0	Opcode Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Opcode Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Opcode Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Opcode Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Subroutine Address Stack Pointer Stack Pointer − 1	1 1 1 0 0	Opcode Irrelevant Data First Subroutine Opcode Return Address (Low Order Byte) Return Address (High Order Byte)

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 2 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Jump Address (High Order Byte)
		3	Opcode Address + 2	1	Jump Address (Low Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Destination Address (High Order Byte)
		3	Opcode Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS LDX LDD	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS STX STD	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR ASR NEG CLR ROL COM ROR DEC TST* INC	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Current Operand Data
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address of Operand	0	New Operand Data
CPX SUBD ADDD	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Operand Address (High Order Byte)
		3	Opcode Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Subroutine (High Order Byte)
		3	Opcode Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

*TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 3 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

*TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

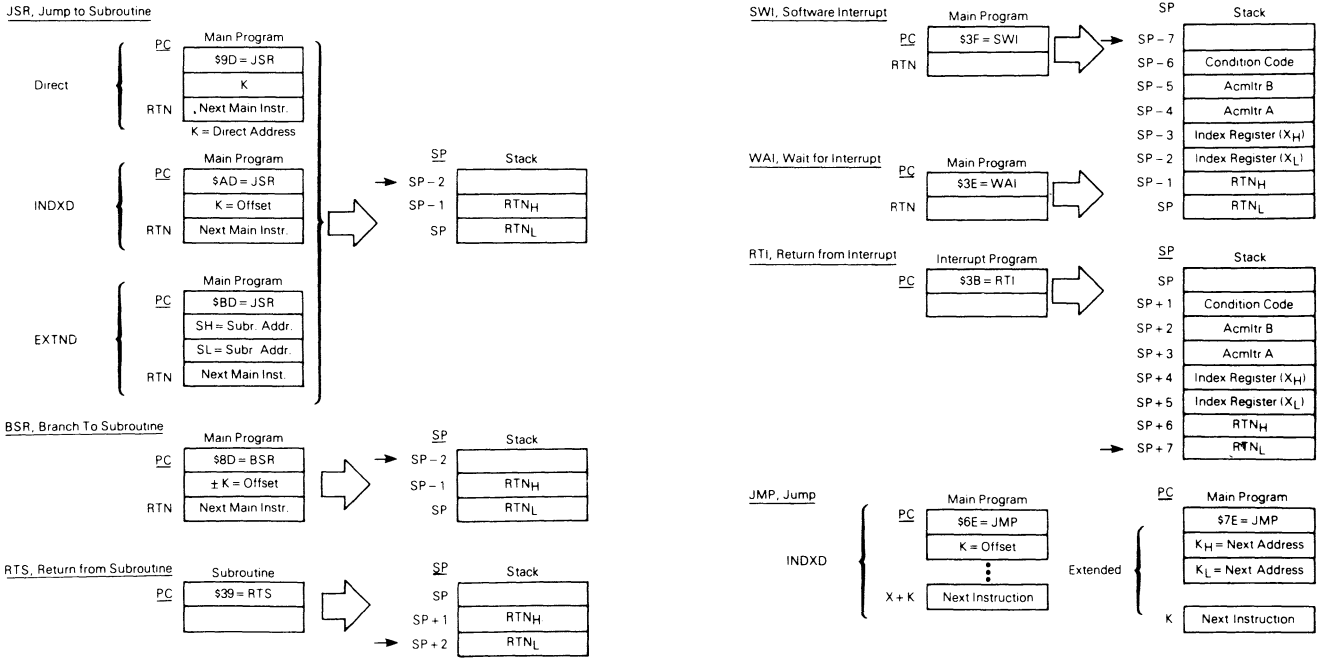
TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1	1 1 0 0	Opcode Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1 Stack Pointer − 2 Stack Pointer − 3 Stack Pointer − 4 Stack Pointer − 5 Stack Pointer − 6	1 1 0 0 0 0 0 0 0	Opcode Opcode of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Condition Code Register

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 5 of 5)

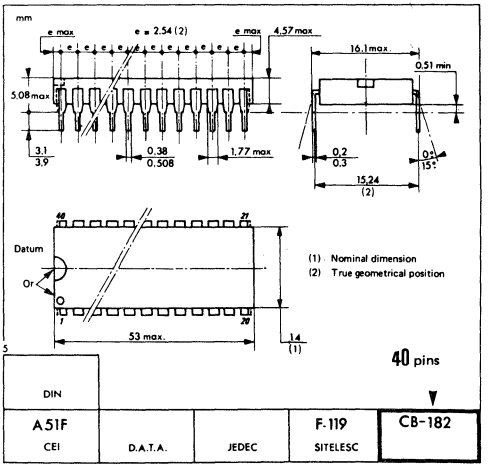
Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
MUL	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer − 1	0	Return Address (High Order Byte)
		5	Stack Pointer − 2	0	Index Register (Low Order Byte)
		6	Stack Pointer − 3	0	Index Register (High Order Byte)
		7	Stack Pointer − 4	0	Contents of Accumulator A
		8	Stack Pointer − 5	0	Contents of Accumulator B
		9	Stack Pointer − 6	0	Contents of Condition Code Register
		10	Stack Pointer − 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMI BVS	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Buss FFFF	1	Low Byte of Restart Vector
BSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

FIGURE 24 — SPECIAL OPERATIONS

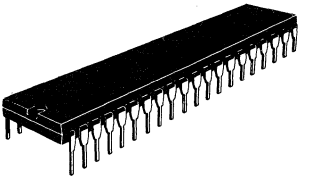


Legend:
RTN = Address of next instruction in Main Program to be executed upon return from subroutine
RTN$_H$ = Most significant byte of Return Address
RTN$_L$ = Least significant byte of Return Address
→ = Stack Pointer After Execution
K = 8-bit Unsigned Value

PHYSICAL DIMENSIONS

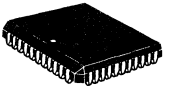


CB-182

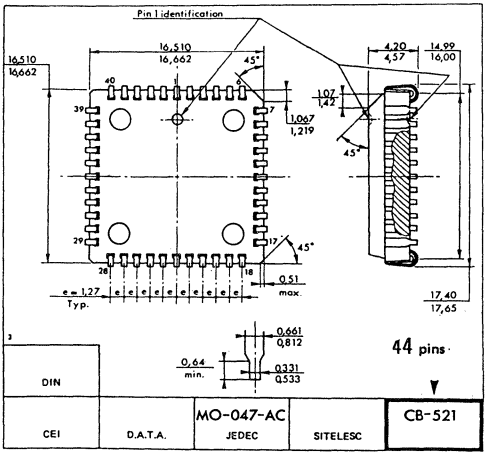


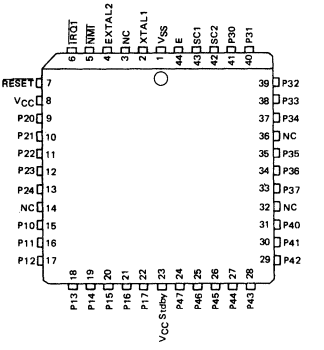
P SUFFIX
PLASTIC PACKAGE

CB-521



FN SUFFIX
PLCC 44





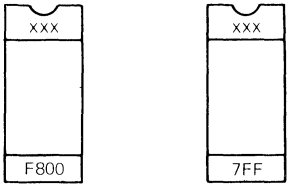
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to THOMSON SEMICONDUCTEURS on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local THOMSON SEMICONDUCTEURS representative or distributor.

EPROMs

Two ET2716 or one ET2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



XXX = Customer ID

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed signed, and returned to THOMSON SEMICONDUCTEURS. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, THOMSON SEMICONDUCTEURS will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by THOMSON SEMICONDUCTEURS. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6801 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from THOMSON SEMICONDUCTEURS factory representatives.

EFDOS is THOMSON SEMICONDUCTEURS' Disk Operating System available on development systems such as DEVICE,...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser,...

*Requires prior factory approval.

Whenever ordering a custom MCU is required, please contact your local THOMSON SEMICONDUCTEURS representative or THOMSON SEMICONDUCTEURS distributor and/or complete and send the attached "MCU customer ordering sheet" to your local THOMSON SEMICONDUCTEURS representative.

ORDERING INFORMATION

EF6801U4 P V

Device — Package — Oper. temp. — Screening level

The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

DEVICE	PACKAGE					OPER. TEMP			SCREENING LEVEL			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6801/03 U4 (1.0 MHz)			●		●	●			●			
			●				●		●			
EF6801/03 U4-1 (1.25 MHz)			●		●	●			●			
			●				●		●			
EF68A01/03 U4 (1.5 MHz)			●			●			●			

Examples : EF6801P, EF6801FN, EF6801PV

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** −40°C to +85°C, **M :** −55°C to +125°C, ***** : may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.

EXORciser is a registered trade mark of MOTOROLA Inc.

THOMSON COMPONENTS MOSTEK

EF6809

8-BIT MICROPROCESSOR UNIT (MPU)

The EF6809 is a revolutionary high-performance 8-bit microprocessor which supports modern programming techniques such as position independence, reentrancy, and modular programming.

This third-generation addition to the 6800 Family has major architectural improvements which include additional registers, instructions, and addressing modes.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The EF6809 has the most complete set of addressing modes available on any 8-bit microprocessor today.

The EF6809 has hardware and software features which make it an ideal processor for higher level language execution or standard controller applications.

EF6800 COMPATIBLE

- Hardware – Interfaces with All 6800 Peripherals
- Software – Upward Source Code Compatible Instruction Set and Addressing Modes

ARCHITECTURAL FEATURES

- Two 16-Bit Index Registers
- Two 16-Bit Indexable Stack Pointers
- Two 8-Bit Accumulators can be Concatenated to Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

HARDWARE FEATURES

- On-Chip Oscillator (Crystal Frequency = 4 × E)
- $\overline{DMA/BREQ}$ Allows DMA Operation on Memory Refresh
- Fast Interrupt Request Input Stacks Only Condition Code Register and Program Counter
- MRDY Input Extends Data Access Times for Use with Slow Memory
- Interrupt Acknowledge Output Allows Vectoring by Devices
- Sync Acknowledge Output Allows for Synchronization to External Event
- Single Bus-Cycle $\overline{RESET}$
- Single 5-Volt Supply Operation
- $\overline{NMI}$ Inhibited After $\overline{RESET}$ Until After First Load of Stack Pointer
- Early Address Valid Allows Use with Slower Memories
- Early Write Data for Dynamic Memories

SOFTWARE FEATURES

- 10 Addressing Modes
  - 6800 Upward Compatible Addressing Modes
  - Direct Addressing Anywhere in Memory Map
  - Long Relative Branches
  - Program Counter Relative
  - True Indirect Addressing
  - Expanded Indexed Addressing:
    - 0-, 5-, 8-, or 16-Bit Constant Offsets
    - 8- or 16-Bit Accumulator Offsets
    - Auto Increment/Decrement by 1 or 2
- Improved Stack Manipulation
- 1464 Instructions with Unique Addressing Modes
- 8 × 8 Unsigned Multiply
- 16-Bit Arithmetic
- Transfer/Exchange All Registers
- Push/Pull Any Registers or Any Set of Registers
- Load Effective Address

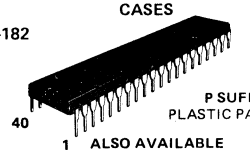
HMOS

(HIGH DENSITY N-CHANNEL, SILICON-GATE)

8-BIT
MICROPROCESSING
UNIT

CASES

CB-182



P SUFFIX
PLASTIC PACKAGE

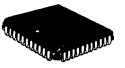
40 1

ALSO AVAILABLE

J SUFFIX
CERDIP PACKAGE

C SUFFIX
CERAMIC PACKAGE

CB-521



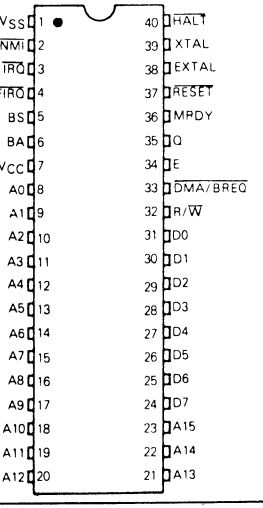
FN SUFFIX
PLCC 44

CB-708

E SUFFIX
LCCC 44

Hi-Rel versions available - See chapter 9

PIN ASSIGNMENT



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range EF6809, EF68A09, EF68B09 EF6809, EF68A09, EF68B09 : V suffix EF6809, EF68A09 : M suffix	T_A	T_L to T_H 0 to +70 −40 to +85 −55 to +125	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage levels (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Cerdip Plastic **PLCC**	θ_{JA}	 50 60 100 **100**	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts − Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts − User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

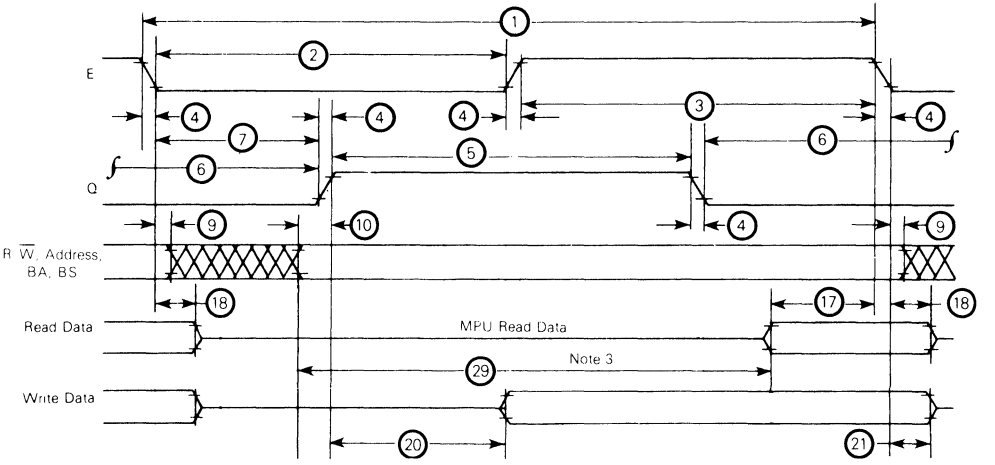
ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 V ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, EXTAL $\overline{RESET}$	V_{IH} V_{IHR}	V_{SS} + 2.0 V_{SS} + 4.0	− −	V_{CC} V_{CC}	V
Input Low Voltage	Logic, EXTAL, $\overline{RESET}$	V_{IL}	V_{SS} − 0.3	−	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V, V_{CC} = max)	Logic	I_{in}	−	−	2.5	μA
dc Output High Voltage (I_{Load} = −205 μA, V_{CC} = min) (I_{Load} = −145 μA, V_{CC} = min) (I_{Load} = −100 μA, V_{CC} = min)	D0-D7 A0-A15, R/$\overline{W}$, Q, E BA, BS	V_{OH}	V_{SS} + 2.4 V_{SS} + 2.4 V_{SS} + 2.4	− − −	− − −	V
dc Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = min)		V_{OL}	−	−	V_{SS} + 0.5	V
Internal Power Dissipation (Measured at T_A = 0°C in Steady State Operation)		P_{INT}	−	−	1.0	W
Capacitance * (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7, $\overline{RESET}$ Logic Inputs, EXTAL, XTAL A0-A15, R/$\overline{W}$, BA, BS	C_{in} C_{out}	− − −	10 10 −	15 15 15	pF pF
Frequency of Operation (Crystal or External Input)	EF6809 EF68A09 EF68B09	f_{XTAL}	0.4 0.4 0.4	− − −	4 6 8	MHz
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = max)	D0-D7 A0-A15, R/$\overline{W}$	I_{TSI}	− −	2.0 −	10 100	μA

* Capacitances are periodically tested rather than 100% tested.

FIGURE 1 – BUS TIMING



BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	EF6809		EF68A09		EF68B09		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time (See Note 5)	t_{cyc}	1.0	10	0.667	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	5000	280	5000	210	5000	ns
3	Pulse Width, E High	PW_{EH}	450	15500	280	15700	220	15700	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
5	Pulse Width, Q High	PW_{QH}	430	5000	280	5000	210	5000	ns
6	Pulse Width, Q Low	PW_{QL}	450	15500	280	15700	220	15700	ns
7	Delay Time, E to Q Rise	t_{AVS}	200	250	130	165	80	125	ns
9	Address Hold Time* (See Note 4)	t_{AH}	20	–	20	–	20	–	ns
10	BA, BS, R/$\overline{W}$, and Address Valid Time to Q Rise	t_{AQ}	50	–	25	–	15	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	60	–	40	–	ns
18	Read Data Hold Time*	t_{DHR}	10	–	10	–	10	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	200	–	140	–	110	ns
21	Write Data Hold Time*	t_{DHW}	30	–	30	–	30	–	ns
29	Usable Access Time (See Note 3)	t_{ACC}	695	–	440	–	330	–	ns
	Processor Control Setup Time (MRDY, Interrupts, $\overline{DMA}$/$\overline{BREQ}$, $\overline{HALT}$, $\overline{RESET}$) (Figures 6, 8, 9, 10, 12, and 13)	t_{PCS}	200	–	140	–	110	–	ns
	Crystal Oscillator Start Time (Figures 6 and 7)	t_{RC}	–	100	–	100	–	100	ms
	Processor Control Rise and Fall Time (Figures 6 and 8)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

*Address and data hold times are periodically tested rather than 100% tested

NOTES:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Usable access time is computed by: 1 – 4 – 7 max + 10 – 17.
4. Hold time (9) for BA and BS is not specified.
5. Maximum t_{cyc} during MRDY or $\overline{DMA}$/$\overline{BREQ}$ is 16 μs.

FIGURE 2 – EF6809 EXPANDED BLOCK DIAGRAM

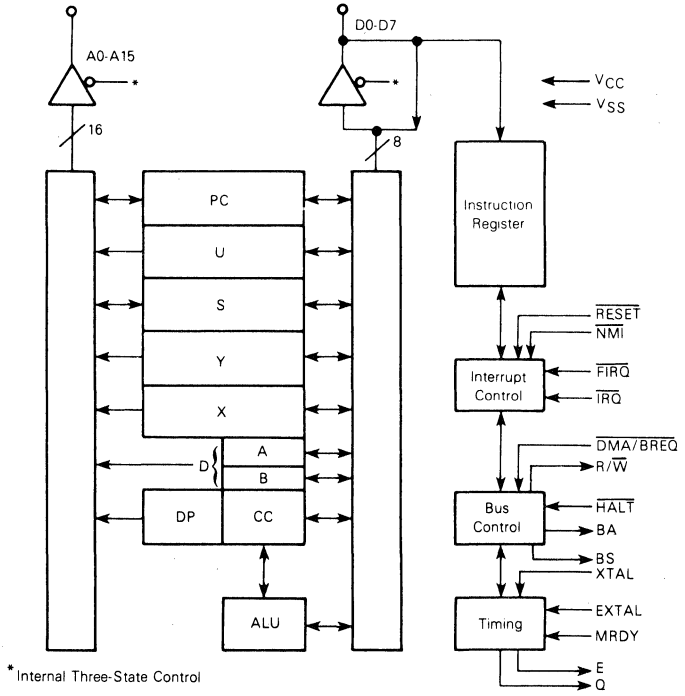
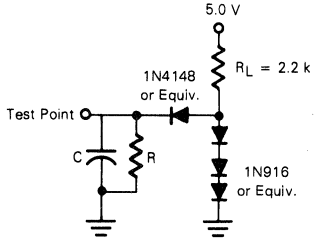


FIGURE 3 – BUS TIMING TEST LOAD



C = 30 pF for BA, BS
130 pF for D0-D7, E, Q
90 pF for A0-A15, R/$\overline{W}$

R = 11.7 kΩ for D0-D7
16.5 kΩ for A0-A15, E, Q, R/$\overline{W}$
24 kΩ for BA, BS

PROGRAMMING MODEL

As shown in Figure 4, the EF6809 adds three registers to the set available in the EF6800. The added registers include a direct page register, the user stack pointer, and a second index register.

ACCUMULATORS (A, B, D)

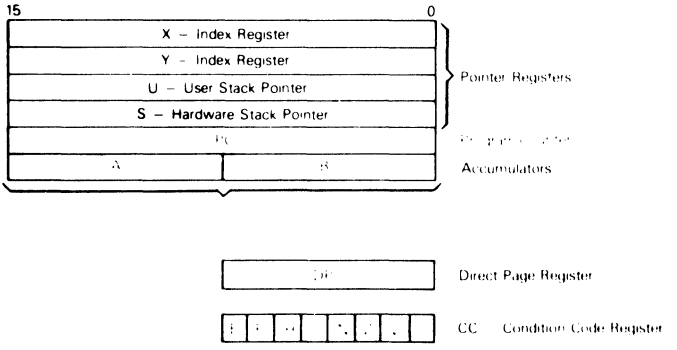
The A and B registers are general purpose accumulators which are used for arithmetic calculations and manipulation of data.

Certain instructions concatenate the A and B registers to form a single 16-bit accumulator. This is referred to as the D register, and is formed with the A register as the most significant byte.

DIRECT PAGE REGISTER (DP)

The direct page register of the EF6809 serves to enhance the direct addressing mode. The content of this register appears at the higher address outputs (A8-A15) during direct addressing instruction execution. This allows the direct mode to be used at any place in memory, under program control. To ensure 6800 compatiblity, all bits of this register are cleared during processor reset.

FIGURE 4 – PROGRAMMING MODEL OF THE MICROPROCESSING UNIT



INDEX REGISTERS (X, Y)

The index registers are used in indexed mode of addressing. The 16-bit address in this register takes part in the calculations of effective addresses. This address may be used to point to data directly or may be modified by an optional constant or register offset. During some indexed modes, the contents of the index register are incremented or decremented to point to the next item of tabular type data. All four pointer registers (X, Y, U, S) may be used as index registers.

STACK POINTER (U,S)

The hardware stack pointer (S) is used automatically by the processor during subroutine calls and interrupts. The stack pointers of the EF6809 point to the top of the stack, in contrast to the EF6800 stack pointer, which pointed to the next free location on the stack. The user stack pointer (U) is controlled exclusively by the programmer. This allows arguments to be passed to and from subroutines with ease. Both stack pointers have the same indexed mode addressing capabilities as the X and Y registers, but also support **Push** and **Pull** instructions. This allows the EF6809 to be used efficiently as a stack processor, greatly enhancing its ability to support higher level languages and modular programming.

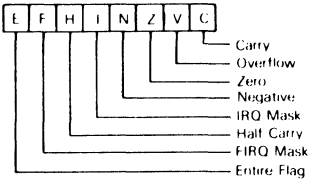
PROGRAM COUNTER

The program counter is used by the processor to point to the address of the next instruction to be executed by the processor. Relative addressing is provided allowing the program counter to be used like an index register in some situations.

CONDITION CODE REGISTER

The condition code register defines the state of the processor at any given time. See Figure 5.

FIGURE 5 CONDITION CODE REGISTER FORMAT



CONDITION CODE REGISTER DESCRIPTION

BIT 0 (C)

Bit 0 is the carry flag, and is usually the carry from the binary ALU. C is also used to represent a "borrow" from subtract like instructions (CMP, NEG, SUB, SBC) and is the complement of the carry from the binary ALU.

BIT 1 (V)

Bit 1 is the overflow flag, and is set to a one by an operation which causes a signed two's complement arithmetic overflow. This overflow is detected in an operation in which the carry from the MSB in the ALU does not match the carry from the MSB-1.

BIT 2 (Z)

Bit 2 is the zero flag, and is set to a one if the result of the previous operation was identically zero.

BIT 3 (N)

Bit 3 is the negative flag, which contains exactly the value of the MSB of the result of the preceding operation. Thus, a negative two's-complement result will leave N set to a one.

BIT 4 (I)

Bit 4 is the $\overline{IRQ}$ mask bit. The processor will not recognize interrupts from the $\overline{IRQ}$ line if this bit is set to a one. $\overline{NMI}$, $\overline{FIRQ}$, $\overline{IRQ}$, $\overline{RESET}$, and SWI all set I to a one. SWI2 and SWI3 do not affect I.

BIT 5 (H)

Bit 5 is the half-carry bit, and is used to indicate a carry from bit 3 in the ALU as a result of an 8-bit addition only (ADC or ADD). This bit is used by the DAA instruction to perform a BCD decimal add adjust operation. The state of this flag is undefined in all subtract-like instructions.

BIT 6 (F)

Bit 6 is the $\overline{FIRQ}$ mask bit. The processor will not recognize interrupts from the $\overline{FIRQ}$ line if this bit is a one. $\overline{NMI}$, $\overline{FIRQ}$, SWI, and $\overline{RESET}$ all set F to a one. $\overline{IRQ}$, SWI2, and SWI3 do not affect F.

BIT 7 (E)

Bit 7 is the entire flag, and when set to a one indicates that the complete machine state (all the registers) was stacked, as opposed to the subset state (PC and CC). The E bit of the stacked CC is used on a return from interrupt (RTI) to determine the extent of the unstacking. Therefore, the current E left in the condition code register represents past action.

PIN DESCRIPTIONS

POWER (V_{SS}, V_{CC})

Two pins are used to supply power to the part: V_{SS} is ground or 0 volts, while V_{CC} is +5.0 V ±5%.

ADDRESS BUS (A0-A15)

Sixteen pins are used to output address information from the MPU onto the address bus. When the processor does not require the bus for a data transfer, it will output address $FFFF_{16}$, R/$\overline{W}$ = 1, and BS = 0; this is a "dummy access" or $\overline{VMA}$ cycle. Addresses are valid on the rising edge of Q. All address bus drivers are made high impedance when output bus available (BA) is high. Each pin will drive one Schottky TTL load or four LSTTL loads, and 90 pF.

DATA BUS (D0-D7)

These eight pins provide communication with the system bidirectional data bus. Each pin will drive one Schottky TTL load or four LSTTL loads, and 130 pF.

READ/WRITE (R/$\overline{W}$)

This signal indicates the direction of data transfer on the data bus. A low indicates that the MPU is writing data onto the data bus. R/$\overline{W}$ is made high impedance when BA is high. R/$\overline{W}$ is valid on the rising edge of Q.

$\overline{RESET}$

A low level on this Schmitt-trigger input for greater than one bus cycle will reset the MPU, as shown in Figure 6. The reset vectors are fetched from locations $FFFE_{16}$ and $FFFF_{16}$ (Table 1) when interrupt acknowledge is true, ($\overline{BA}$•BS = 1). During initial power on, the $\overline{RESET}$ line should be held low until the clock oscillator is fully operational. See Figure 7.

Because the EF6809 $\overline{RESET}$ pin has a Schmitt-trigger input with a threshold voltage higher than that of standard peripherals, a simple R/C network may be used to reset the entire system. This higher threshold voltage ensures that all peripherals are out of the reset state before the processor

$\overline{HALT}$

A low level on this input pin will cause the MPU to stop running at the end of the present instruction and remain halted indefinitely without loss of data. When halted, the BA output is driven high indicating the buses are high impedance. BS is also high which indicates the processor is in the halt or bus grant state. While halted, the MPU will not respond to external real-time requests ($\overline{FIRQ}$, $\overline{IRQ}$) although $\overline{DMA/BREQ}$ will always be accepted, and $\overline{NMI}$ or $\overline{RESET}$ will be latched for later response. During the halt state, Q and E continue to run normally. If the MPU is not running ($\overline{RESET}$, $\overline{DMA/BREQ}$), a halted state (BA•BS = 1) can be achieved by pulling $\overline{HALT}$ low while $\overline{RESET}$ is still low. If $\overline{DMA/BREQ}$ and $\overline{HALT}$ are both pulled low, the processor will reach the last cycle of the instruction (by reverse cycle stealing) where the machine will the become halted. See Figure 8.

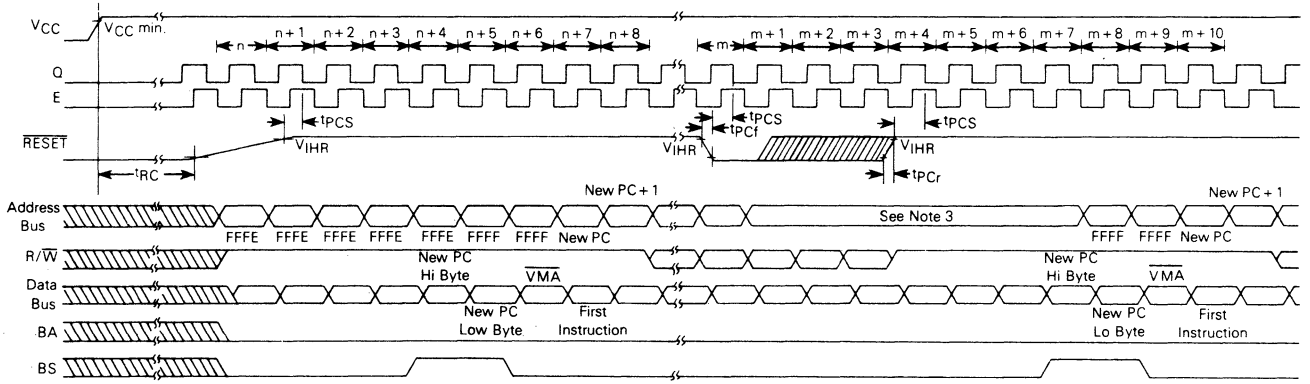
BUS AVAILABLE, BUS STATUS (BA, BS)

The bus available output is an indication of an internal control signal which makes the MOS buses of the MPU high impedance. This signal does not imply that the bus will be available for more than one cycle. When BA goes low, a dead cycle will elapse before the MPU acquires the bus.

The bus status output signal, when decoded with BA, represents the MPU state (valid with leading edge of Q).

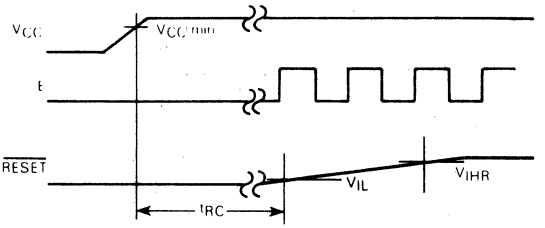
MPU State		MPU State Definition
BA	BS	
0	0	Normal (Running)
0	1	Interrupt or Reset Acknowledge
1	0	Sync Acknowledge
1	1	Halt or Bus Grant Acknowledge

FIGURE 6 — $\overline{\text{RESET}}$ TIMING



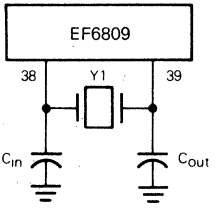
NOTES: 1. Parts with date codes prefixed by 7F or 5A will come out of $\overline{\text{RESET}}$ one cycle sooner than shown.
2. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.
3. FFFE appears on the bus during $\overline{\text{RESET}}$ low time. Following the active transition of the $\overline{\text{RESET}}$ line, three more FFFE cycles will appear followed by the vector fetch.

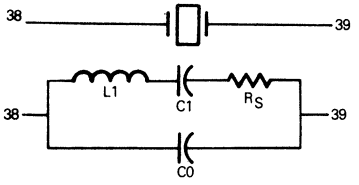
FIGURE 7 – CRYSTAL CONNECTIONS AND OSCILLATOR START UP



NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

Y1	C_{in}	C_{out}
8 MHz	18 pF	18 pF
6 MHz	20 pF	20 pF
4 MHz	24 pF	24 pF





Nominal Crystal Parameters

	3.58 MHz	4.00 MHz	6.0 MHz	8.0 MHz
R_S	60 Ω	50 Ω	30-50 Ω	20-40 Ω
C0	3.5 pF	6.5 pF	4-6 pF	4-6 pF
C1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF
Q	>40 k	>30 k	>20 k	>20 k

All parameters are 10%

NOTE: These are representative AT-cut crystal parameters only. Crystals of other types of cut may also be used.

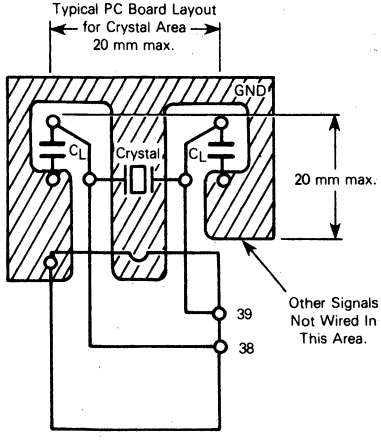
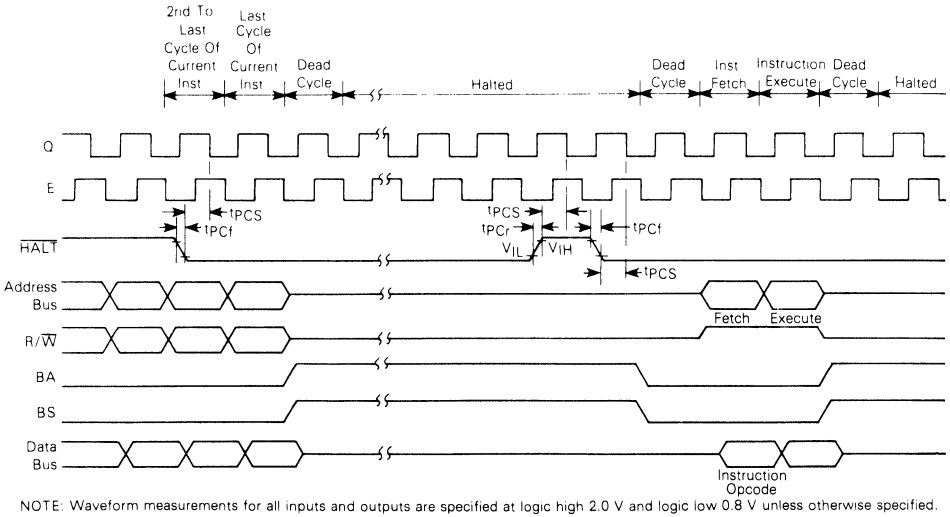


FIGURE 8 – $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION FOR SYSTEM DEBUG



NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

INTERRUPT ACKNOWLEDGE is indicated during both cycles of a hardware-vector-fetch ($\overline{\text{RESET}}$, $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$, SWI, SWI2, SWI3). This signal, plus decoding of the lower four address lines, can provide the user with an indication of which interrupt level is being serviced and allow vectoring by device. See Table 1.

SYNC ACKNOWLEDGE is indicated while the MPU is waiting for external synchronization on an interrupt line.

HALT/BUS GRANT is true when the MC6809 is in a halt or bus grant condition.

TABLE 1 – MEMORY MAP FOR INTERRUPT VECTORS

Memory Map For Vector Locations		Interrupt Vector Description
MS	LS	
FFFE	FFFF	$\overline{\text{RESET}}$
FFFC	FFFD	$\overline{\text{NMI}}$
FFFA	FFFB	SWI
FFF8	FFF9	$\overline{\text{IRQ}}$
FFF6	FFF7	$\overline{\text{FIRQ}}$
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reserved

NON MASKABLE INTERRUPT ($\overline{\text{NMI}}$)*

A negative transition on this input requests that a non-maskable interrupt sequence be generated. A non-maskable interrupt cannot be inhibited by the program, and also has a higher priority than $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$, or software interrupts. During recognition of an $\overline{\text{NMI}}$, the entire machine state is saved on the hardware stack. After reset, an $\overline{\text{NMI}}$ will not be recognized until the first program load of the hardware stack pointer (S). The pulse width of $\overline{\text{NMI}}$ low must be at least one E cycle. If the $\overline{\text{NMI}}$ input does not meet the minimum set up with respect to Q, the interrupt will not be recognized until the next cycle. See Figure 9.

FAST-INTERRUPT REQUEST ($\overline{\text{FIRQ}}$)*

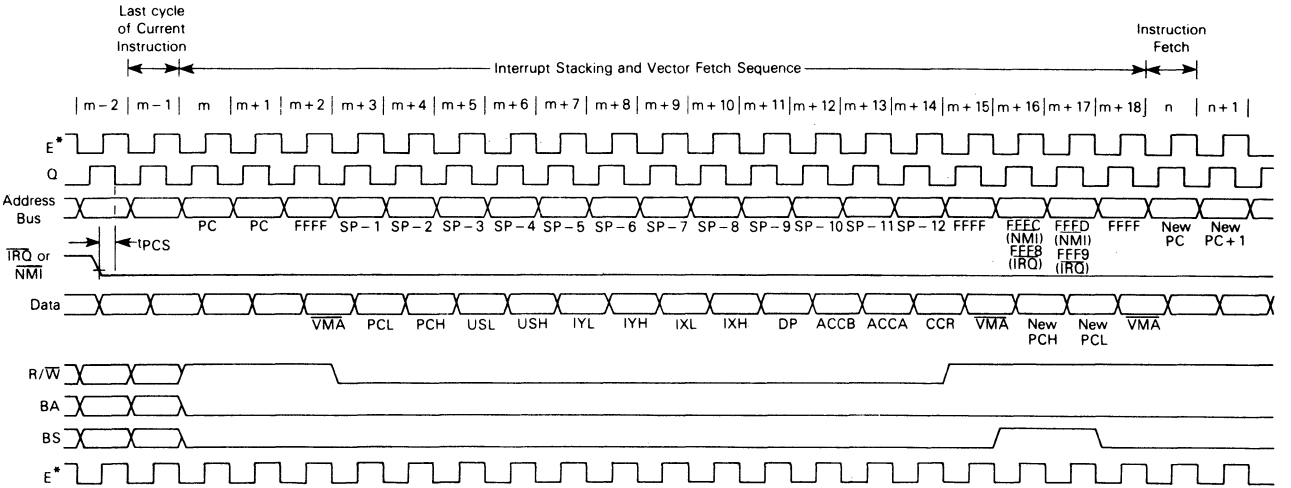
A low level on this input pin will initiate a fast interrupt sequence, provided its mask bit (F) in the CC is clear. This sequence has priority over the standard interrupt request ($\overline{\text{IRQ}}$), and is fast in the sense that it stacks only the contents of the condition code register and the program counter. The interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 10.

INTERRUPT REQUEST ($\overline{\text{IRQ}}$)*

A low level input on this pin will initiate an interrupt request sequence provided the mask bit (I) in the CC is clear. Since $\overline{\text{IRQ}}$ stacks the entire machine state it provides a slower response to interrupts than $\overline{\text{FIRQ}}$. $\overline{\text{IRQ}}$ also has a lower priority than $\overline{\text{FIRQ}}$. Again, the interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 9.

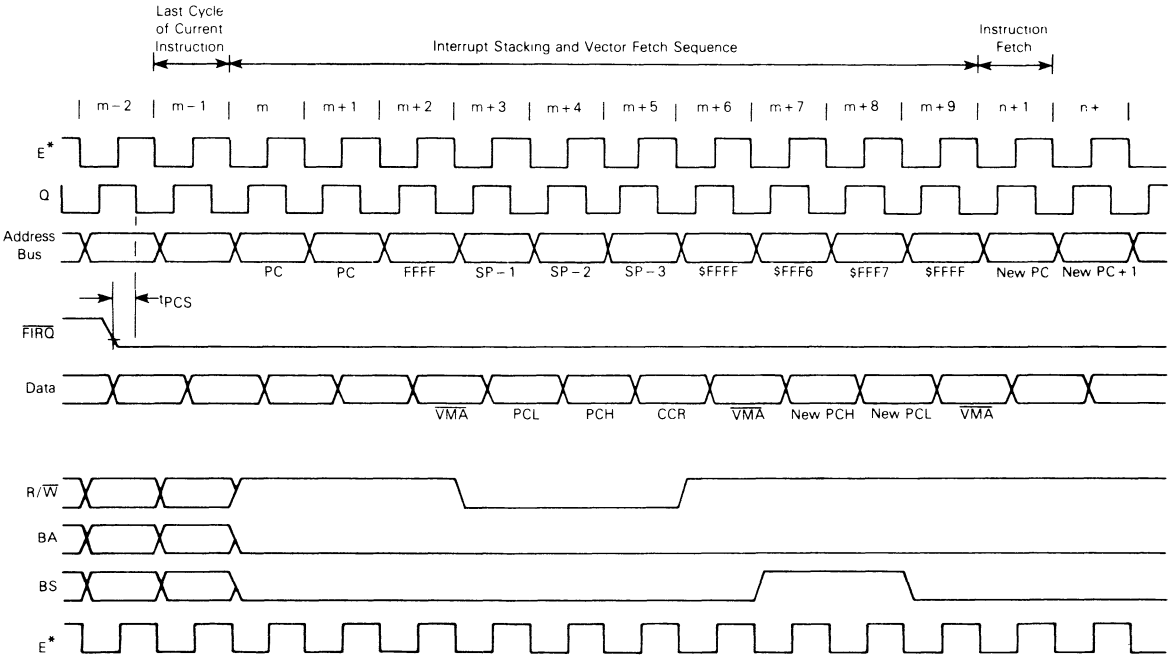
*$\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, and $\overline{\text{IRQ}}$ requests are sampled on the falling edge of Q. One cycle is required for synchronization before these interrupts are recognized. The pending interrupt(s) will not be serviced until completion of the current instruction unless a SYNC or CWAI condition is present. If $\overline{\text{IRQ}}$ and $\overline{\text{FIRQ}}$ do not remain low until completion of the current instruction they may not be recognized. However, $\overline{\text{NMI}}$ is latched and need only remain low for one cycle. No interrupts are recognized or latched between the falling edge of $\overline{\text{RESET}}$ and the rising edge of BS indicating $\overline{\text{RESET}}$ acknowledge.

FIGURE 9 — $\overline{IRQ}$ AND $\overline{NMI}$ INTERRUPT TIMING



NOTE: Waveform measurements for all inputs and outputs are specified at logic high = 2.0 V and logic low = 0.8 V unless otherwise specified.
*E clock shown for reference only.

FIGURE 10 – $\overline{\text{FIRQ}}$ INTERRUPT TIMING



NOTE: Waveform measurements for all inputs and outputs are specified at logic high = 2.0 V and logic low = 0.8 V unless otherwise specified.
*E clock shown for reference only.

XTAL, EXTAL

These inputs are used to connect the on-chip oscillator to an external parallel-resonant crystal. Alternately, the pin EXTAL may be used as a TTL level input for external timing by grounding XTAL. The crystal or external frequency is four times the bus frequency. See Figure 7. Proper RF layout techniques should be observed in the layout of printed circuit boards.

E, Q

E is similar to the EF6800 bus timing signal phase 2; Q is a quadrature clock signal which leads E. Q has no parrallel on the EF6800. Addresses from the MPU will be valid with the leading edge of Q. Data is latched on the falling edge of E. Timing for E and Q is shown in Figure 11.

MRDY*

This input control signal allows stretching of E and Q to extend data-access time. E and Q operate normally while MRDY is high. When MRDY is low, E and Q may be stretched in integral multiples of quarter (¼) bus cycles, thus allowing interface to slow memories, as shown in Figure 12(a). During non-valid memory access ($\overline{VMA}$ cycles), MRDY has no effect on stretching E and Q; this inhibits slowing the processor during "don't care" bus accesses. MRDY may also be used to stretch clocks (for slow memory) when bus control has been transferred to an external device (through the use of $\overline{HALT}$ and $\overline{DMA/BREQ}$).

$\overline{DMA/BREQ}$*

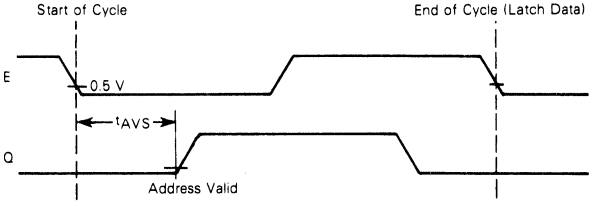
The $\overline{DMA/BREQ}$ input provides a method of suspending execution and acquiring the MPU bus for another use, as shown in Figure 13. Typical uses include DMA and dynamic memory refresh.

A low level on this pin will stop instruction execution at the end of the current cycle unless pre-empted by self-refresh. The MPU will acknowledge $\overline{DMA/BREQ}$ by setting BA and BS to a one. The requesting device will now have up to 15 bus cycles before the MPU retrieves the bus for self-refresh. Self-refresh requires one bus cycle with a leading and trailing dead cycle. See Figure 14. The self-refresh counter is only cleared if $\overline{DMA/BREQ}$ is inactive for two or more MPU cycles.

Typically, the DMA controller will request to use the bus by asserting $\overline{DMA/BREQ}$ pin low on the leading edge of E. When the MPU replies by setting BA and BS to a one, that cycle will be a dead cycle used to transfer bus mastership to the DMA controller.

False memory accesses may be prevented during any dead cycles by developing a system $\overline{DMAVMA}$ signal which is LOW in any cycle when BA has changed.

FIGURE 11 — E/Q RELATIONSHIP



NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

* The on-board clock generator furnishes E and Q to both the system and the MPU. When MRDY is pulled low, both the system clocks and the internal MPU clocks are stretched. Assertion of $\overline{DMA/BREQ}$ input stops the internal MPU clocks while allowing the external system clocks to RUN (i.e., release the bus to a DMA controller). The internal MPU clocks resume operation after $\overline{DMA/BREQ}$ is released or after 16 bus cycles (14 DMA, two dead), whichever occurs first. While $\overline{DMA/BREQ}$ is asserted it is sometimes necessary to pull MRDY low to allow DMA to/from slow memory/peripherals. As both MRDY and $\overline{DMA/BREQ}$ control the internal MPU clocks, care must be exercised not to violate the maximum t_{cyc} specification for MRDY or $\overline{DMA/BREQ}$. (Maximum t_{cyc} during MRDY or $\overline{DMA/BREQ}$ is 16 μs.)

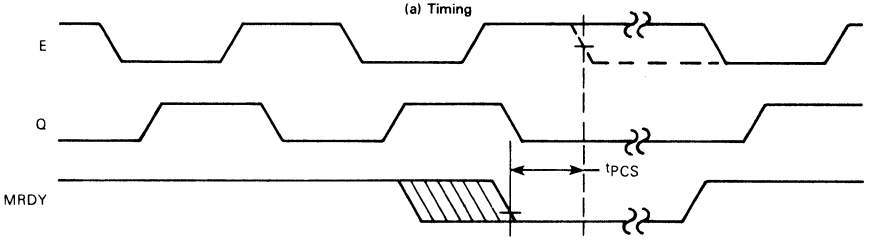
When BA goes low (either as a result of $\overline{DMA/BREQ}$ = HIGH or MPU self-refresh), the DMA device should be taken off the bus. Another dead cycle will elapse before the MPU accesses memory to allow transfer of bus mastership without contention.

MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function. This sequence begins after $\overline{RESET}$ and is repeated indefinitely unless altered by a special instruction or hardware occurrence. Software instructions that alter normal MPU operation are: SWI, SWI2, SWI3, CWAI, RTI, and SYNC. An interrupt, $\overline{HALT}$, or $\overline{DMA/BREQ}$ can also alter the normal execution of instructions. Figure 15 illustrates the flowchart for the EF6809.

FIGURE 12 – MRDY TIMING AND SYNCHRONIZATION

(a) Timing



(b) Synchronization

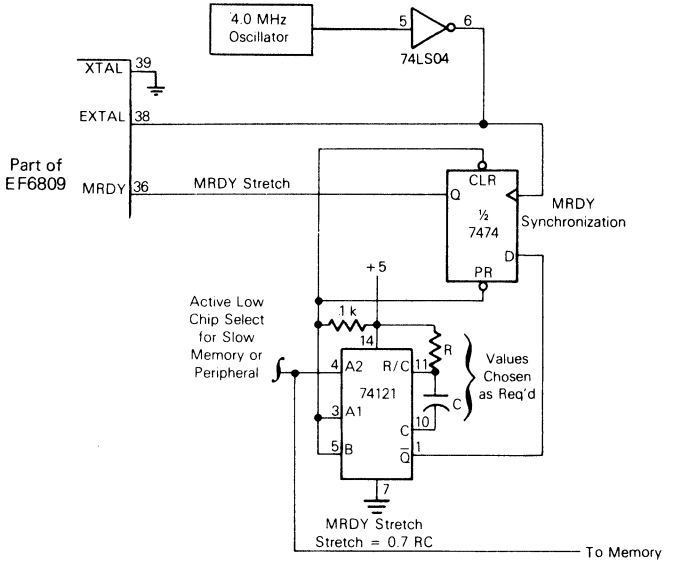


FIGURE 13 – TYPICAL DMA TIMING (< 14 CYCLES)

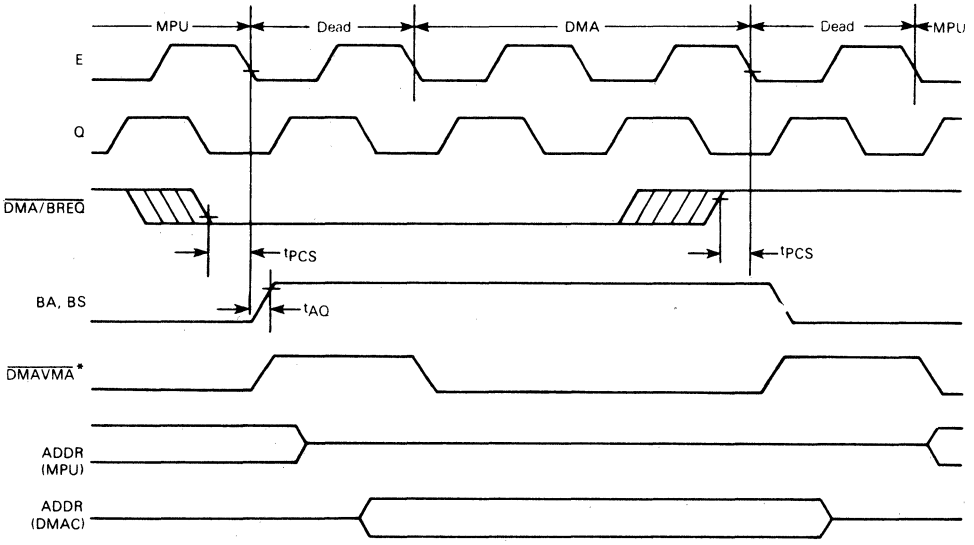
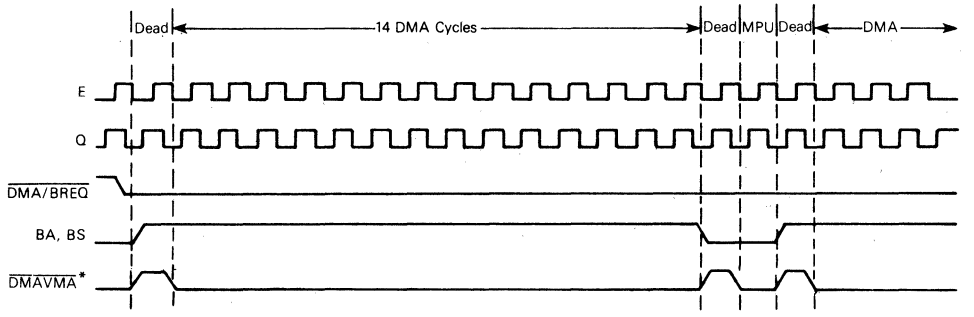
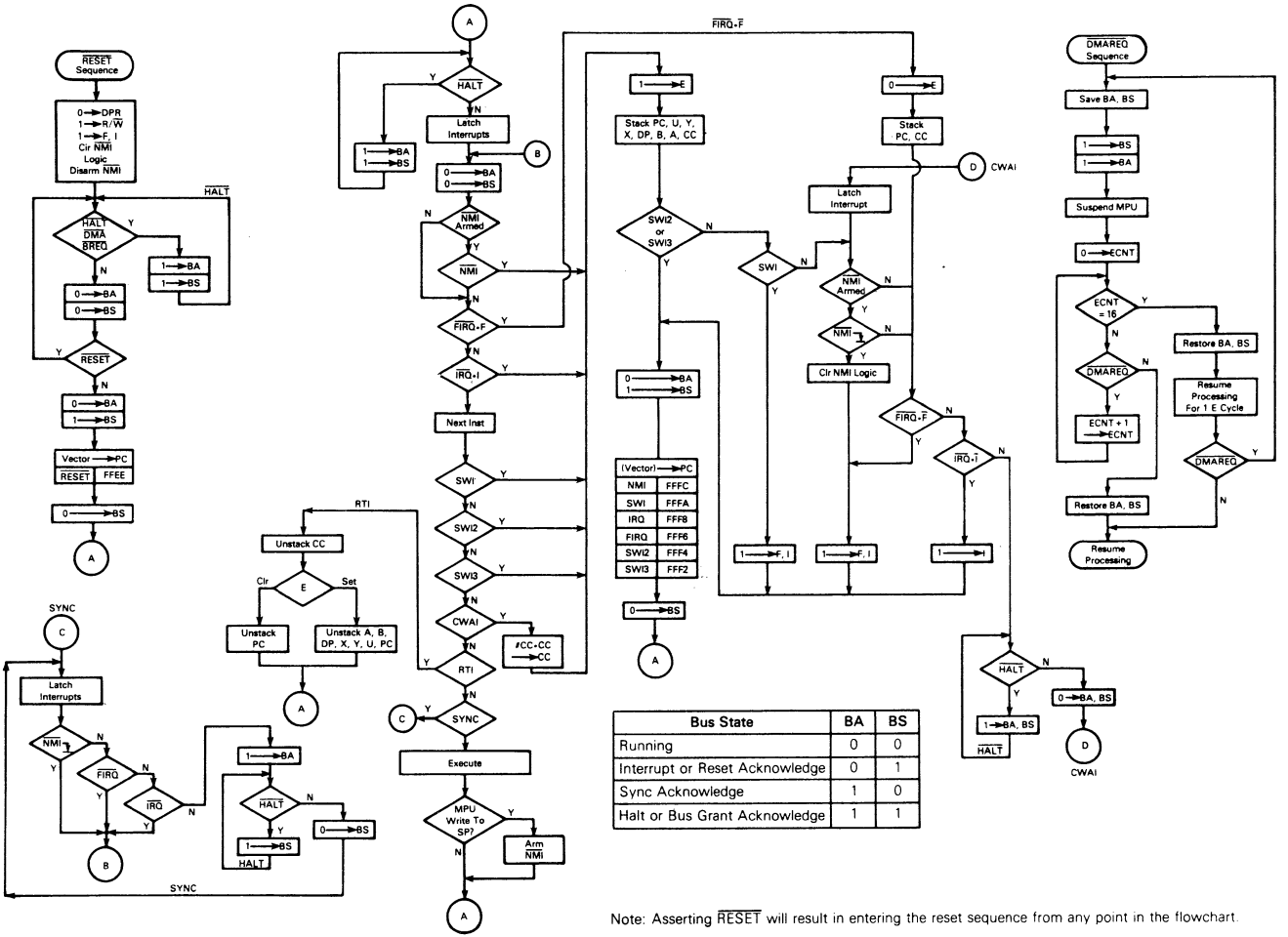


FIGURE 14 – AUTO-REFRESH DMA TIMING (> 14 CYCLES)
(REVERSE CYCLE STEALING)



* DMAVMA is a signal which is developed externally, but is a system requirement for DMA.

NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.



Bus State	BA	BS
Running	0	0
Interrupt or Reset Acknowledge	0	1
Sync Acknowledge	1	0
Halt or Bus Grant Acknowledge	1	1

Note: Asserting $\overline{\text{RESET}}$ will result in entering the reset sequence from any point in the flowchart.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The EF6809 has the most complete set of addressing modes available on any microcomputer today. For example, the EF6809 has 59 basic instructions; however, it recognizes 1464 different variations of instructions and addressing modes. The addressing modes support modern programming techniques. The following addressing modes are available on the EF6809 :

- Inherent (includes accumulator)
- Immediate
- Extended
  - Extended Indirect
- Direct
- Register
- Indexed
  - Zero-Offset
  - Constant Offset
  - Accumulator Offset
  - Auto Increment/Decrement
  - Indexed Indirect
- Relative
  - Short/Long Relative Branching
  - Program Counter Relative Addressing

INHERENT (INCLUDES ACCUMULATOR)

In this addressing mode, the opcode of the instruction contains all the address information necessary. Examples of inherent addressing are: ABX, DAA, SWI, ASRA, and CLRB.

IMMEDIATE ADDRESSING

In immediate addressing, the effective address of the data is the location immediately following the opcode (i.e., the data to be used in the instruction immediately following the opcode of the instruction). The EF6809 uses both 8- and 16-bit immediate values depending on the size of argument specified by the opcode. Examples of instructions with immediate addressing are:

```
LDA #$20
LDX #$F000
LDY #CAT
```

NOTE

signifies Immediate addressing; $ signifies hexadecimal value.

EXTENDED ADDRESSING

In extended addressing, the contents of the two bytes immediately following the opcode fully specify the 16-bit effective address used by the instruction. Note that the address generated by an extended instruction defines an absolute address and is not position independent. Examples of extended addressing include:

```
LDA   CAT
STX   MOUSE
LDD   $2000
```

EXTENDED INDIRECT – As in the special case of indexed addressing (discussed below), one level of indirection may be added to extended addressing. In extended indirect, the two bytes following the postbyte of an indexed instruction contain the address of the data.

```
LDA   [CAT]
LDX   [$FFFE]
STU   [DOG]
```

DIRECT ADDRESSING

Direct addressing is similar to extended addressing except that only one byte of address follows the opcode. This byte specifies the lower eight bits of the address to be used. The upper eight bits of the address are supplied by the direct page register. Since only one byte of address is required in direct addressing, this mode requires less memory and executes faster than extended addressing. Of course, only 256 locations (one page) can be accessed without redefining the contents of the DP register. Since the DP register is set to $00 on reset, direct addressing on the EF6809 is compatible with direct addressing on the 6800. Indirection is not allowed in direct addressing. Some examples of direct addressing are:

```
LDA   $30
SETDP $10 (assembler directive)
LDB   $1030
LDD   < CAT
```

NOTE

< is an assembler directive which forces direct addressing.

REGISTER ADDRESSING

Some opcodes are followed by a byte that defines a register or set of registers to be used by the instruction. This is called a postbyte. Some examples of register addressing are:

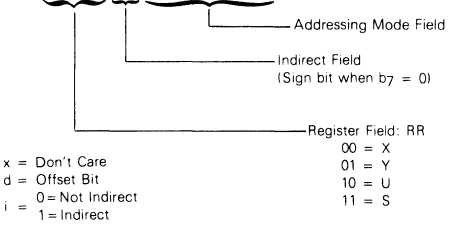
```
TFR   X, Y          Transfers X into Y
EXG   A, B          Exchanges A with B
PSHS  A, B, X, Y    Push Y, X, B and A onto S
PULU  X, Y, D       Pull D, X, and Y from U
```

INDEXED ADDRESSING

In all indexed addressing, one of the pointer registers (X, Y, U, S, and sometimes PC) is used in a calculation of the effective address of the operand to be used by the instruction. Five basic types of indexing are available and are discussed below. The postbyte of an indexed instruction specifies the basic type and variation of the addressing mode as well as the pointer register to be used. Figure 16 lists the legal formats for the postbyte. Table 2 gives the assembler form and the number of cycles and bytes added to the basic values for indexed addressing for each variation.

FIGURE 16 — INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

Postbyte Register Bit								Indexed Addressing Mode
7	6	5	4	3	2	1	0	
0	R	R	d	d	d	d	d	EA = ,R + 5 Bit Offset
1	R	R	0	0	0	0	0	,R+
1	R	R	i	0	0	0	1	,R+ +
1	R	R	0	0	0	1	0	,– R
1	R	R	i	0	0	1	1	,– – R
1	R	R	i	0	1	0	0	EA = ,R + 0 Offset
1	R	R	i	0	1	0	1	EA = ,R + ACCB Offset
1	R	R	i	0	1	1	0	EA = ,R + ACCA Offset
1	R	R	i	1	0	0	0	EA = ,R + 8 Bit Offset
1	R	R	i	1	0	0	1	EA = ,R + 16 Bit Offset
1	R	R	i	1	0	1	1	EA = ,R + D Offset
1	x	x	i	1	1	0	0	EA = ,PC + 8 Bit Offset
1	x	x	i	1	1	0	1	EA = ,PC + 16 Bit Offset
1	R	R	i	1	1	1	1	EA = [,Address]



ZERO-OFFSET INDEXED – In this mode, the selected pointer register contains the effective address of the data to be used by the instruction. This is the fastest indexing mode.

Examples are:

```
LDD   0,X
LDA   S
```

CONSTANT OFFSET INDEXED – In this mode, a two's complement offset and the contents of one of the pointer registers are added to form the effective address of the operand. The pointer register's initial content is unchanged by the addition.

Three sizes of offsets are available:

5 bit (– 16 to + 15)
8 bit (– 128 to + 127)
16 bit (– 32768 to + 32767)

The two's complement 5-bit offset is included in the postbyte and, therefore, is most efficient in use of bytes and cycles. The two's complement 8-bit offset is contained in a single byte following the postbyte. The twos complement 16-bit offset is in the two bytes following the postbyte. In most cases the programmer need not be concerned with the size of this offset since the assembler will select the optimal size automatically.

Examples of constant-offset indexing are:

```
LDA   23,X
LDX   – 2,S
LDY   300,X
LDU   CAT,Y
```

TABLE 2 — INDEXED ADDRESSING MODE

Type	Forms	Non Indirect		+ ~	+ #	Indirect		+ ~	+ #
		Assembler Form	Postbyte Opcode			Assembler Form	Postbyte Opcode		
Constant Offset From R	No Offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
(2s Complement Offsets)	5-Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8-Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16-Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R (2s Complement Offsets)	A Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
	B Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	,R+	1RR00000	2	0	not allowed			
	Increment By 2	,R+ +	1RR00001	3	0	[,R+ +]	1RR10001	6	0
	Decrement By 1	,– R	1RR00010	2	0	not allowed			
	Decrement By 2	,– – R	1RR00011	3	0	[,– – R]	1RR10011	6	0
Constant Offset From PC (2s Complement Offsets)	8-Bit Offset	n, PCR	1xx01100	1	1	[n, PCR]	1xx11100	4	1
	16-Bit Offset	n, PCR	1xx01101	5	2	[n, PCR]	1xx11101	8	2
Extended Indirect	16-Bit Address	–	–	–	–	[n]	10011111	5	2

R = X, Y, U, or S
x = Don't Care

RR:
00 = X
01 = Y
10 = U
11 = S

$^{+}_{\sim}$ and $^{+}_{\#}$ indicate the number of additional cycles and bytes for the particular variation.

ACCUMULATOR-OFFSET INDEXED This mode is similar to constant offset indexed except that the two's complement value in one of the accumulators (A, B, or D) and the contents of one of the pointer registers are added to form the effective address of the operand. The contents of both the accumulator and the pointer register are unchanged by the addition. The postbyte specifies which accumulator to use as an offset and no additional bytes are required. The advantage of an accumulator offset is that the value of the offset can be calculated by a program at run-time.

Some examples are:

```
LDA   B,Y
LDX   D,Y
LEAX  B,X
```

AUTO INCREMENT/DECREMENT INDEXED – In the auto increment addressing mode, the pointer register contains the address of the operand. Then, after the pointer register is used it is incremented by one or two. This addressing mode is useful in stepping through tables, moving data, or for the creation of software stacks. In auto decrement, the pointer register is decremented prior to use as the address of the data. The use of auto decrement is similar to that of auto increment; but the tables, etc., are scanned from the high to low addresses. The size of the increment/ decrement can be either one or two to allow for tables of either 8- or 16-bit data to be accessed and is selectable by the programmer. The pre-decrement, post-increment nature of these modes allows them to be used to create additional software stacks that behave identically to the U and S stacks.

Some examples of the auto increment/decrement addressing modes are:

```
LDA   ,X+
STD   ,Y++
LDB   ,-Y
LDX   ,--S
```

Care should be taken in performing operations on 16-bit pointer registers (X, Y, U, S) where the same register is used to calculate the effective address.

Consider the following instruction:

STX 0,X++ (X initialized to 0)

The desired result is to store zero in locations $0000 and $0001 then increment X to point to $0002. In reality, the following occurs:

```
0→temp        calculate the EA; temp is a holding register
X+2→X         perform auto increment
X→(temp)      do store operation
```

INDEXED INDIRECT – All of the indexing modes, with the exception of auto increment/decrement by one or a ±4-bit offset, may have an additional level of indirection specified. In indirect addressing, the effective address is contained at the location specified by the contents of the index register plus any offset. In the example below, the A accumulator is loaded indirectly using an effective address calculated from the index register and an offset.

```
          Before Execution
          A = XX (don't care)
          X = $F000
$0100     LDA  [$10,X]     EA is now $F010

$F010     $F1              $F150 is now the
$F011     $50              new EA

$F150     $AA
          After Execution
          A = $AA Actual Data Loaded
          X = $F000
```

All modes of indexed indirect are included except those which are meaningless (e.g., auto increment/decrement by one indirect). Some examples of indexed indirect are:

```
LDA   [,X]
LDD   [10,S]
LDA   [B,Y]
LDD   [,X++]
```

RELATIVE ADDRESSING

The byte(s) following the branch opcode is (are) treated as a signed offset which may be added to the program counter. If the branch condition is true, then the calculated address (PC + signed offset) is loaded into the program counter. Program execution continues at the new location as indicated by the PC; short (one byte offset) and long (two bytes offset) relative addressing modes are available. All of memory can be reached in long relative addressing as an effective address is interpreted modulo 2^{16}. Some examples of relative addressing are:

```
          BEQ     CAT      (short)
          BGT     DOG      (short)
CAT       LBEQ    RAT      (long)
DOG       LBGT    RABBIT   (long)
          •
          •
          •
RAT       NOP
RABBIT    NOP
```

PROGRAM COUNTER RELATIVE – The PC can be used as the pointer register with 8- or 16-bit signed offsets. As in relative addressing, the offset is added to the current PC to create the effective address. The effective address is then used as the address of the operand or data. Program counter relative addressing is used for writing position independent programs. Tables related to a particular routine will maintain the same relationship after the routine is moved, if referenced relative to the program counter. Examples are:

```
LDA   CAT, PCR
LEAX  TABLE, PCR
```

Since program counter relative is a type of indexing, an additional level of indirection is available.

```
LDA   [CAT, PCR]
LDU   [DOG, PCR]
```

The instruction set of the EF6809E is similar to that of the 6800 and is upward compatible at the source code level. The number of opcodes has been reduced from 72 to 59, but because of the expanded architecture and additional addressing modes, the number of available opcodes (with different addressing modes) has risen from 197 to 1464.

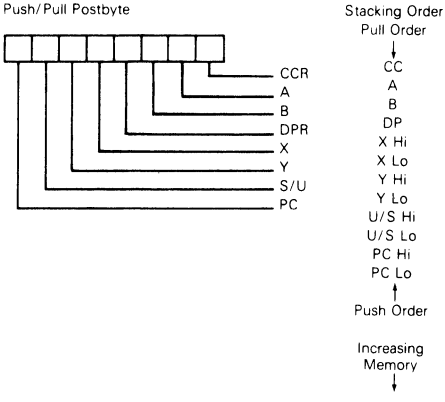
Some of the new instructions are described in detail below.

PSHU/PSHS

The push instructions have the capability of pushing onto either the hardware stack (S) or user stack (U) any single register or set of registers with a single instruction.

PULU/PULS

The pull instructions have the same capability of the push instruction, in reverse order. The byte immediately following the push or pull opcode determines which register or registers are to be pushed or pulled. The actual push/pull sequence is fixed; each bit defines a unique register to push or pull, as shown below.



TFR/EXG

Within the EF6809E, any register may be transferred to or exchanged with another of like size, i.e., 8 bit to 8 bit or 16 bit to 16 bit. Bits 4-7 of postbyte define the source register, while bits 0-3 represent the destination register. These are denoted as follows:

Transfer/Exchange Postbyte

Source	Destination

Register Field

0000 = D (A:B)	1000 = A
0001 = X	1001 = B
0010 = Y	1010 = CCR
0011 = U	1011 = DPR
0100 = S	
0101 = PC	

NOTE

All other combinations are undefined and INVALID.

LEAX/LEAY/LEAU/LEAS

The LEA (load effective address) works by calculating the effective address used in an indexed instruction and stores that address value, rather than the data at that address, in a pointer register. This makes all the features of the internal addressing hardware available to the programmer. Some of the implications of this instruction are illustrated in Table 3.

The LEA instruction also allows the user to access data and tables in a position independent manner. For example:

```
        LEAX    MSG1, PCR
        LBSR    PDATA (print message routine)
        •
        •
MSG1    FCC     'MESSAGE'
```

This sample program prints: 'MESSAGE'. By writing MSG1, PCR, the assembler computes the distance between the present address and MSG1. This result is placed as a constant into the LEAX instruction which will be indexed from the PC value at the time of execution. No matter where the code is located when it is executed, the computed offset from the PC will put the absolute address of MSG1 into the X pointer register. This code is totally position independent.

The LEA instructions are very powerful and use an internal holding register (temp). Care must be exercised when using the LEA instructions with the auto increment and auto decrement addressing modes due to the sequence of internal operations. The LEA internal sequence is outlined as follows:

LEAa ,b+ (any of the 16-bit pointer registers X, Y, U, or S may be substituted for a and b)

1. b → temp (calculate the EA)
2. b + 1 → b (modify b, postincrement)
3. temp → a (load a)

LEAa , – b

1. b – 1 → temp (calculate EA with predecrement)
2. b – 1 → b (modify b, predecrement)
3. temp → a (load a)

TABLE 3 — LEA EXAMPLES

Instruction	Operation	Comment
LEAX 10, X	X + 10 → X	Adds 5-Bit Constant 10 to X
LEAX 500, X	X + 500 → X	Adds 16-Bit Constant 500 to X
LEAY A, Y	Y + A → Y	Adds 8-Bit A Accumulator to Y
LEAY D, Y	Y + D → Y	Adds 16-Bit D Accumulator to Y
LEAU – 10, U	U – 10 → U	Substracts 10 from U
LEAS – 10, S	S – 10 → S	Used to Reserve Area on Stack
LEAS 10, S	S + 10 → S	Used to 'Clean Up' Stack
LEAX 5, S	S + 5 → X	Transfers As Well As Adds

Auto increment-by-two and auto decrement-by-two instructions work similarly. Note that LEAX ;X+ does not change X; however, LEAX, −X does decrement; LEAX 1, X should be used to increment X by one.

MUL

Multiplies the unsigned binary numbers in the A and B accumulator and places the unsigned result into the 16-bit D accumulator. The unsigned multiply also allows multiple-precision multiplications.

LONG AND SHORT RELATIVE BRANCHES

The EF6809 has the capability of program counter relative branching throughout the entire memory map. In this mode, if the branch is to be taken, the 8- or 16-bit signed offset is added to the value of the program counter to be used as the effective address. This allows the program to branch anywhere in the 64K memory map. Position-independent code can be easily generated through the use of relative branching. Both short (8-bit) and long (16-bit) branches are available.

SYNC

After encountering a sync instruction, the MPU enters a sync state, stops processing instructions, and waits for an interrupt. If the pending interrupt is non-maskable ($\overline{NMI}$) or maskable ($\overline{FIRQ}$, $\overline{IRQ}$) with its mask bit (F or I) clear, the processor will clear the sync state and perform the normal interrupt stacking and service routine. Since $\overline{FIRQ}$ and $\overline{IRQ}$ are not edge-triggered, a low level with a minimum duration of three bus cycles is required to assure that the interrupt will be taken. If the pending interrupt is maskable ($\overline{FIRQ}$, $\overline{IRQ}$) with its mask bit (F or I) set, the processor will clear the sync state and continue processing by executing the next in-line instruction. Figure 17 depicts sync timing.

SOFTWARE INTERRUPTS

A software interrupt is an instruction which will cause an interrupt and its associated vector fetch. These software interrupts are useful in operating system calls, software debugging, trace operations, memory mapping, and software development systems. Three levels of SWI are available on the EF6809, and are prioritized in the following order: SWI, SWI2, SWI3.

16-BIT OPERATION

The EF6809, has the capability of processing 16-bit data. These instructions include loads, stores, compares, adds, subtracts, transfers, exchanges, pushes, and pulls.

CYCLE-BY-CYCLE OPERATION

The address bus cycle-by-cycle performance chart (Figure 18) illustrates the memory-access sequence corresponding to each possible instruction and addressing mode in the EF6809. Each instruction begins with an opcode fetch. While that opcode is being internally decoded, the next program byte is always fetched. (Most instructions will use the next byte, so this technique considerably speeds throughput.) Next, the operation of each opcode will follow the flowchart. $\overline{VMA}$ is an indication of $FFFF_{16}$ on the address bus, $R/\overline{W} = 1$ and $BS = 0$. The following examples illustrate the use of the chart.

Example 1: LBSR (Branch Taken)
Before Execution SP = F000

```
                  •
                  •
                  •
$8000             LBSR   CAT
                  •
                  •
                  •
$A000  CAT        •
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	R/$\overline{W}$	Description
1	8000	17	1	Opcode Fetch
2	8001	20	1	Offset High Byte
3	8002	00	1	Offset Low Byte
4	FFFF	*	1	$\overline{VMA}$ Cycle
5	FFFF	*	1	$\overline{VMA}$ Cycle
6	A000	*	1	Computed Branch Address
7	FFFF	*	1	$\overline{VMA}$ Cycle
8	EFFF	80	0	Stack High Order Byte of Return Address
9	EFFE	03	0	Stack Low Order Byte of Return Address

Example 2: DEC (Extended)

```
$8000   DEC    $A000
          •
          •
          •
$A8000  $80
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	R/$\overline{W}$	Description
1	8000	7A	1	Opcode Fetch
2	8001	A0	1	Operand Address, High Byte
3	8002	00	1	Operand Address, Low Byte
4	FFFF	*	1	$\overline{VMA}$ Cycle
5	A000	80	1	Read the Data
6	FFFF	*	1	$\overline{VMA}$ Cycle
7	A000	7F	0	Store the Decremented Data

*The data bus has the data at that particular address.

INSTRUCTION SET TABLES

The instructions of the EF6809 have been broken down into five different categories. They are as follows:

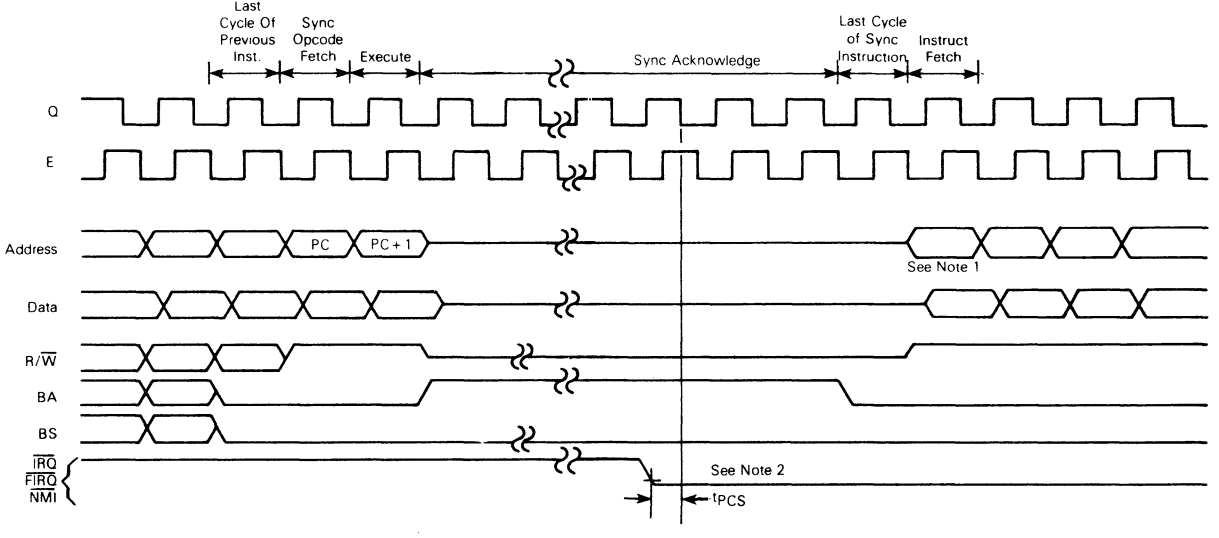
8-bit operation (Table 4)
16-bit operation (Table 5)
Index register/stack pointer instructions (Table 6)
Relative branches (long or short) (Table 7)
Miscellaneous instructions (Table 8)

Hexadecimal values for the instructions are given in Table 9.

PROGRAMMING AID

Figure 19 contains a compilation of data that will assist in programming the EF6809.

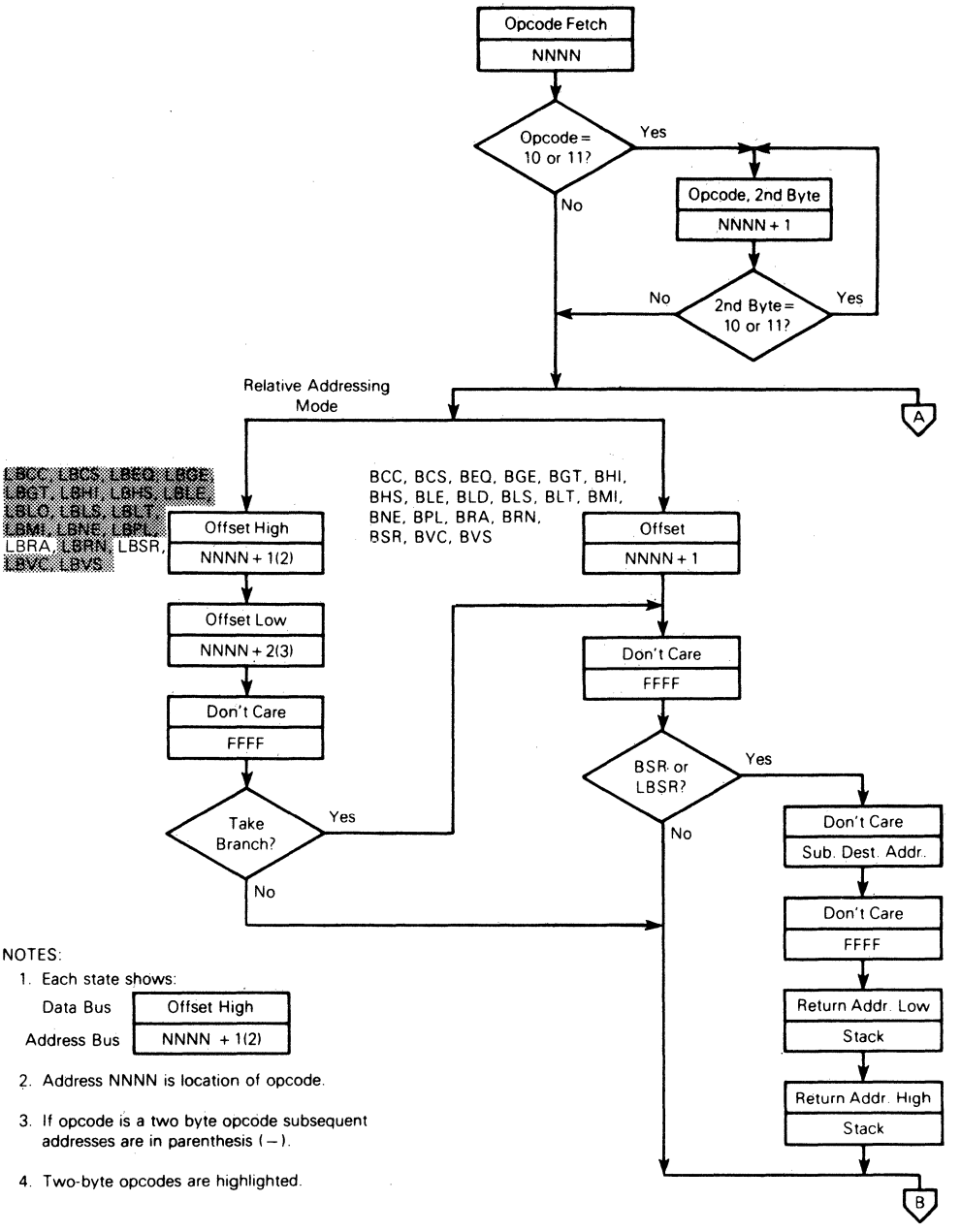
FIGURE 17 — SYNC TIMING



NOTES:

1. If the associated mask bit is set when the interrupt is requested, this cycle will be an instruction fetch from address location PC + 1. However, if the interrupt is accepted (NMI or an unmasked FIRQ or IRQ) interrupt processing continues with this cycle as m on Figures 9 and 10 (Interrupt Timing).
2. If mask bits are clear, IRQ and FIRQ must be held low for three cycles to guarantee interrupt to be taken, although only one cycle is necessary to bring the processor out of SYNC.
3. Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 1 of 9)



NOTES:

1. Each state shows:

Data Bus	Offset High
Address Bus	NNNN + 1(2)

2. Address NNNN is location of opcode.
3. If opcode is a two byte opcode subsequent addresses are in parenthesis (–).
4. Two-byte opcodes are highlighted.

FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 2 of 9)

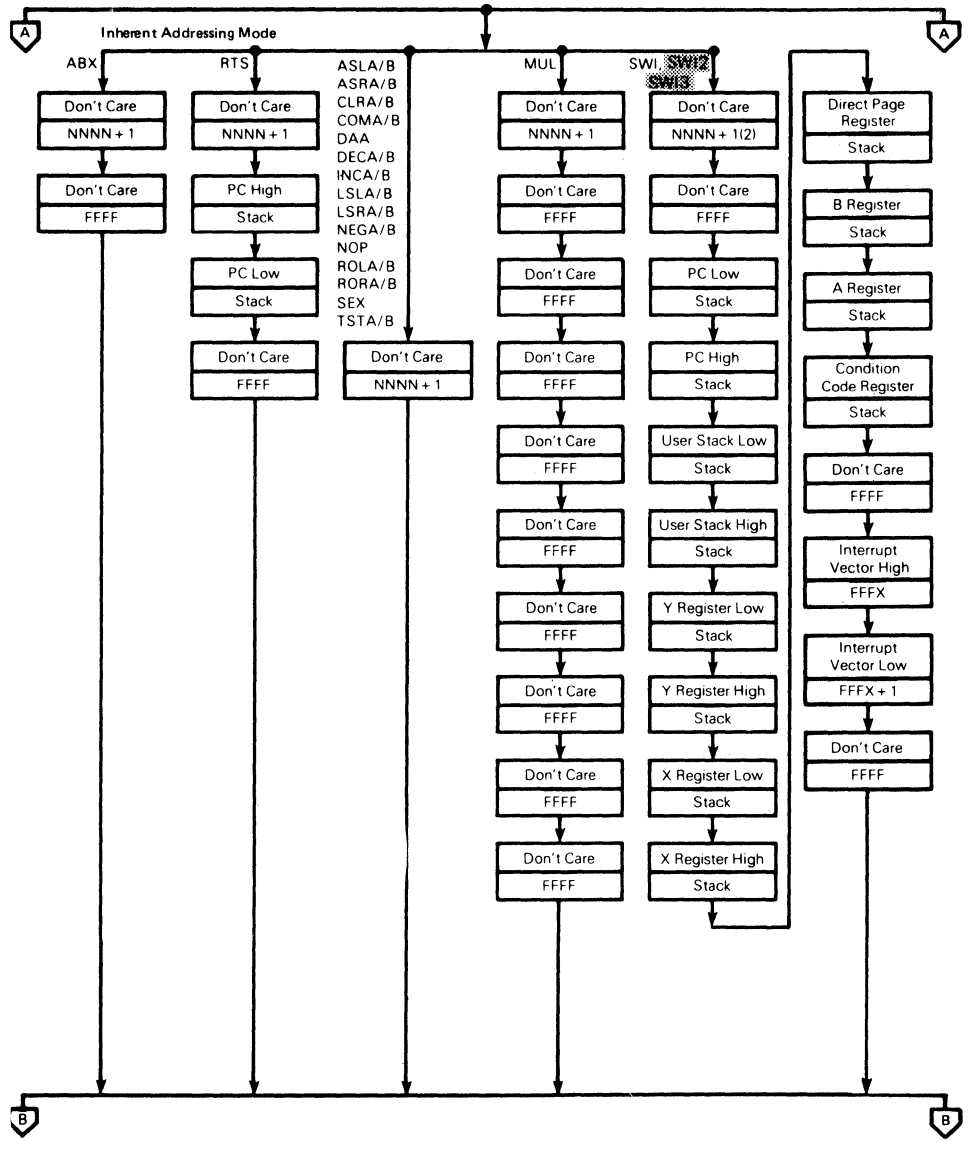


FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 3 of 9)

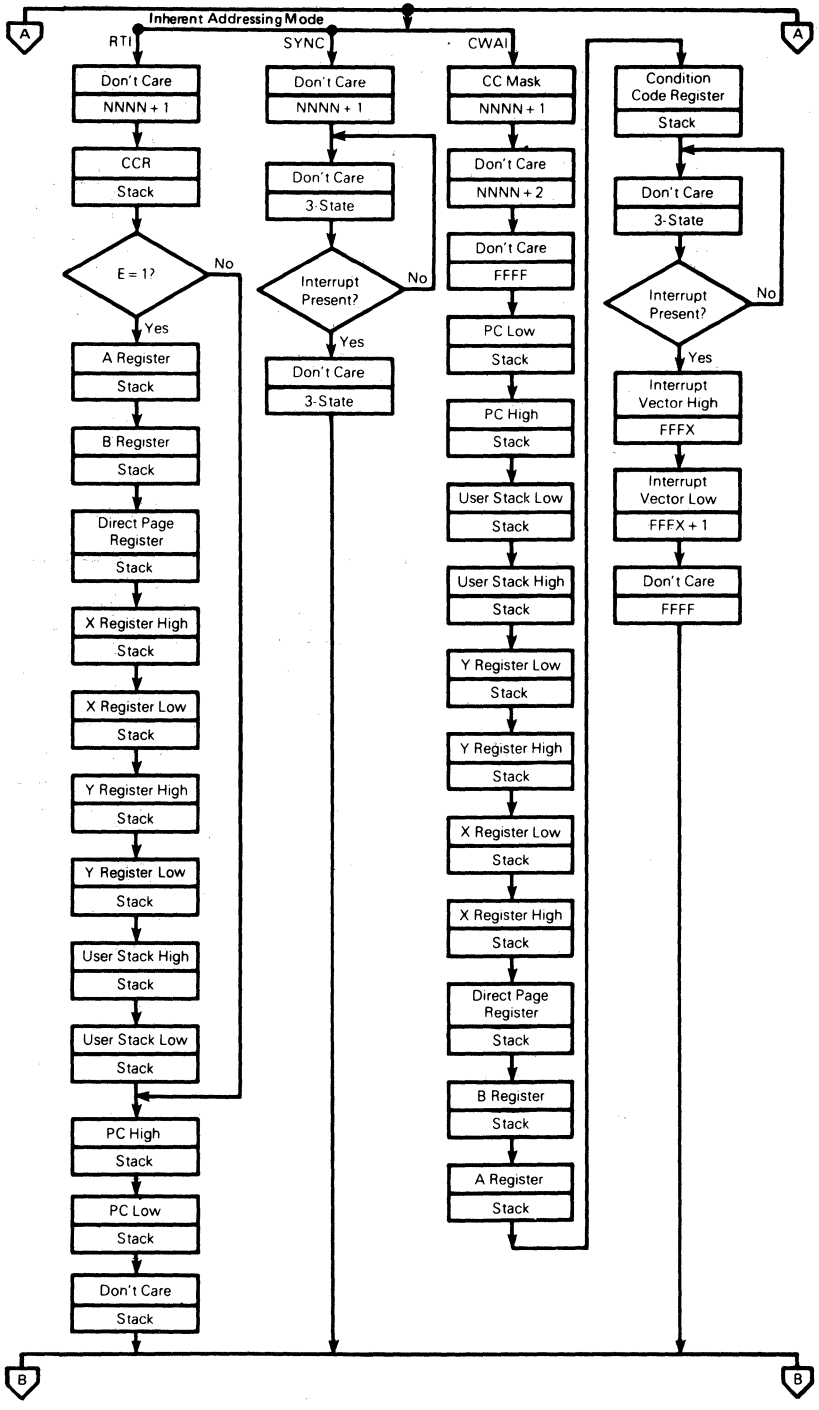


FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 4 of 9)

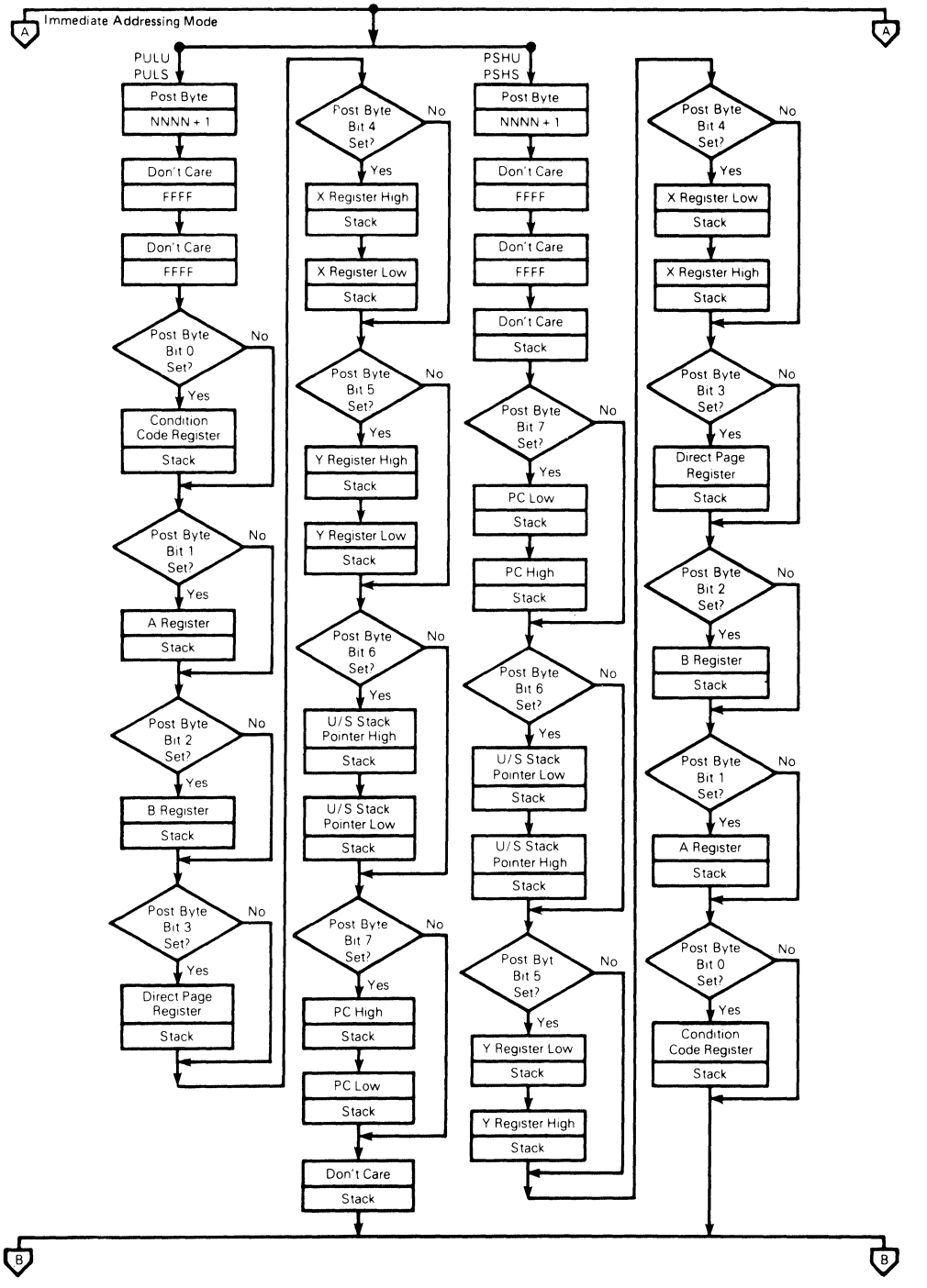


FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 5 of 9)

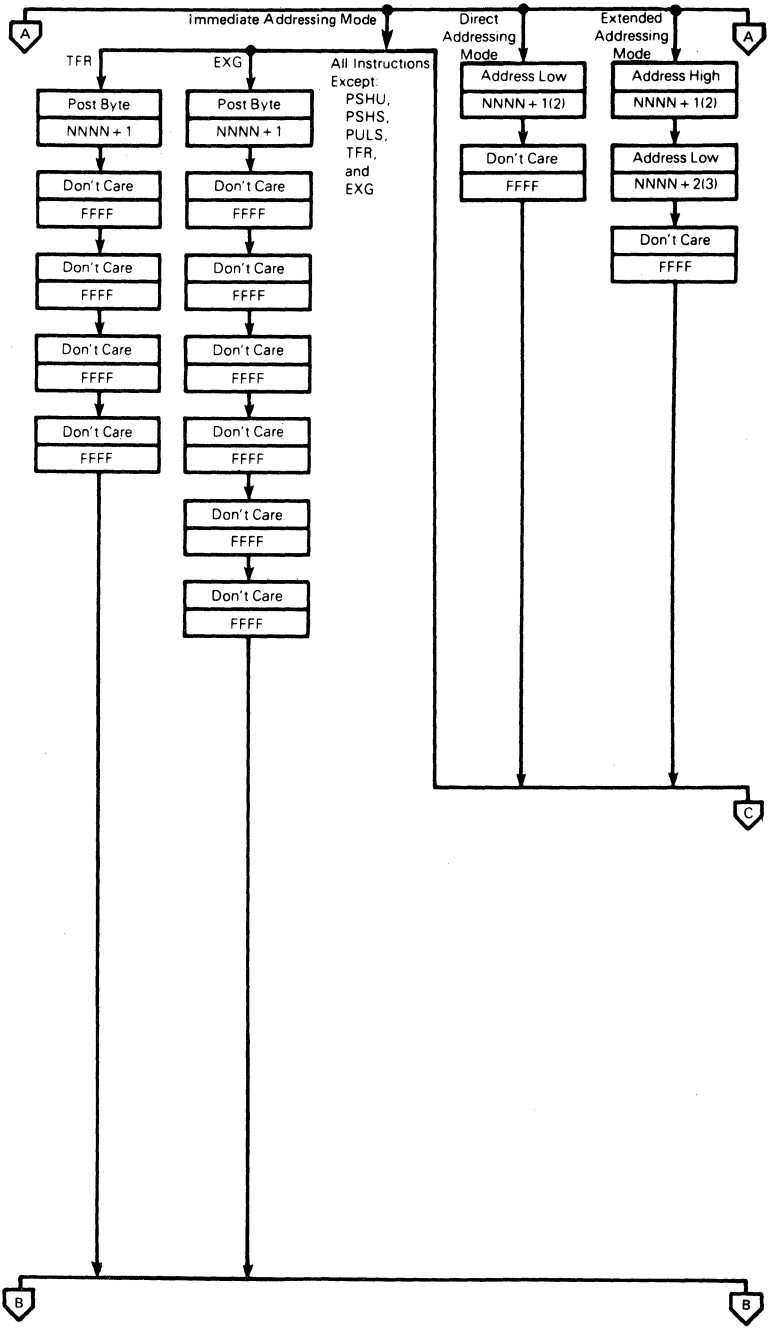
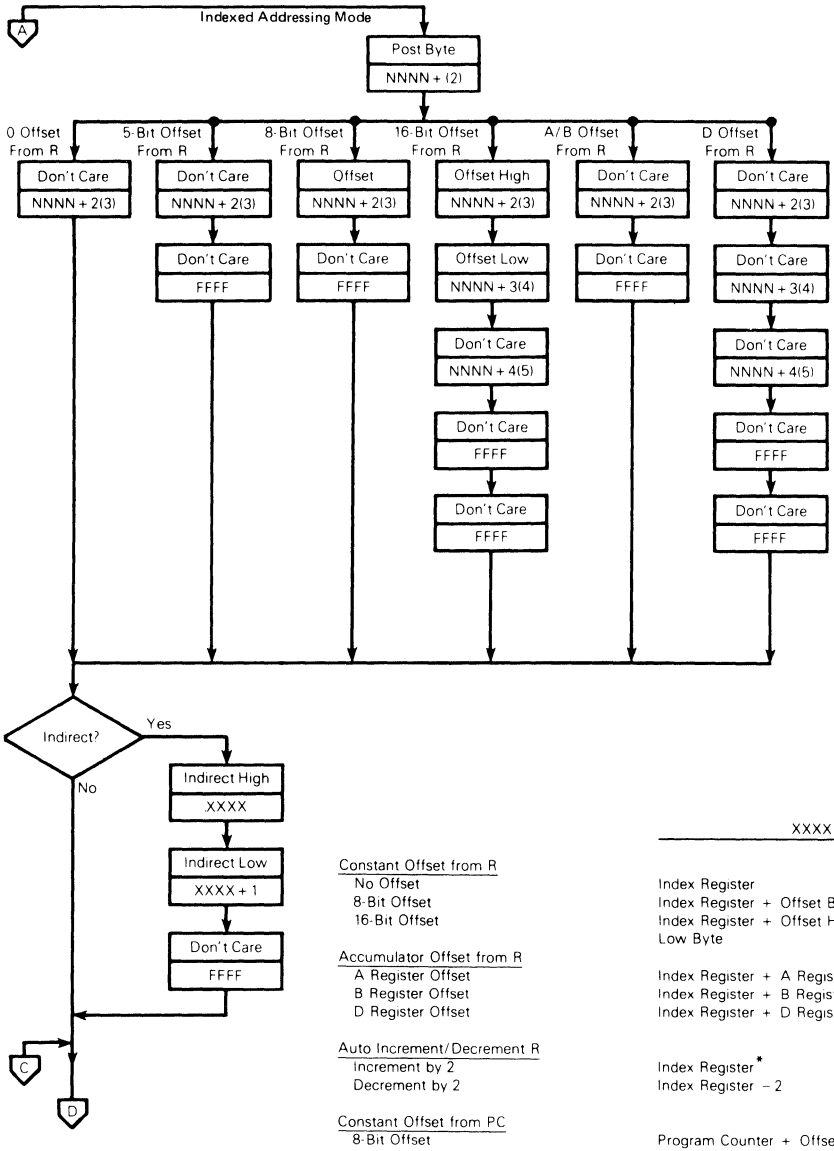


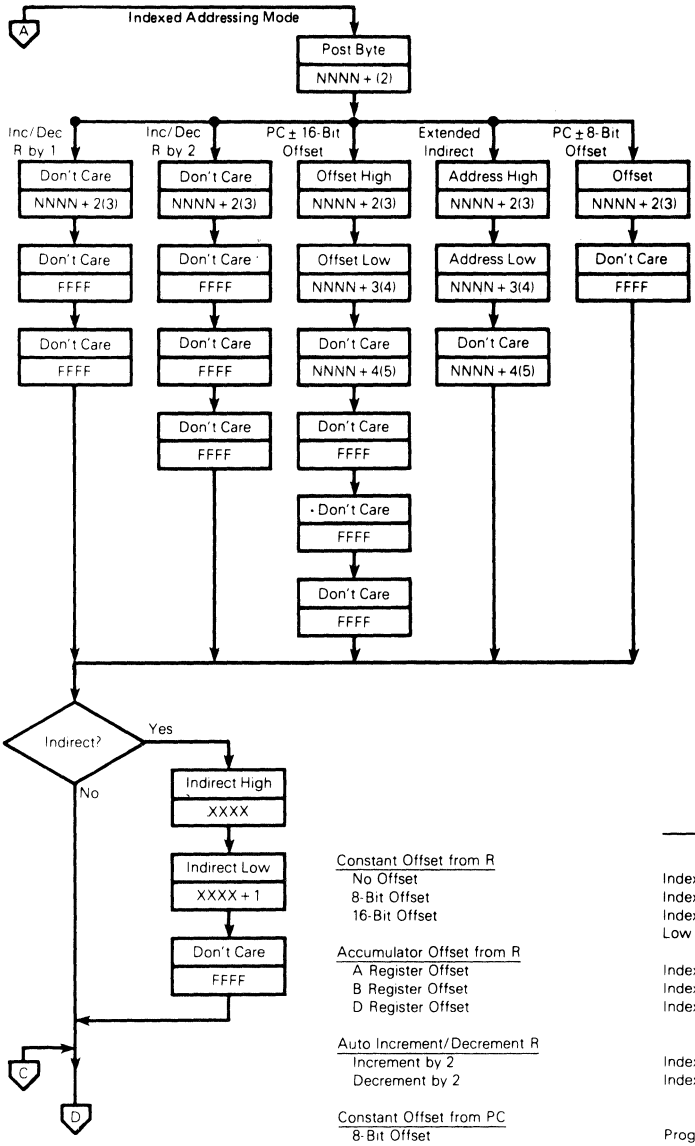
FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 6 of 9)



	XXXX
Constant Offset from R	
No Offset	Index Register
8-Bit Offset	Index Register + Offset Byte
16-Bit Offset	Index Register + Offset High Byte: Offset Low Byte
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 2	Index Register*
Decrement by 2	Index Register – 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-bit Offset	Program Counter + Offset High Byte Offset Low Byte
Extended Indirect	
16-Bit Address	Address High Byte Addres Low Byte

* The index register is incremented following the indexed access

FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 7 of 9)



Constant Offset from R
- No Offset — Index Register
- 8-Bit Offset — Index Register + Offset Byte
- 16-Bit Offset — Index Register + Offset High Byte: Offset Low Byte

Accumulator Offset from R
- A Register Offset — Index Register + A Register
- B Register Offset — Index Register + B Register
- D Register Offset — Index Register + D Register

Auto Increment/Decrement R
- Increment by 2 — Index Register*
- Decrement by 2 — Index Register – 2

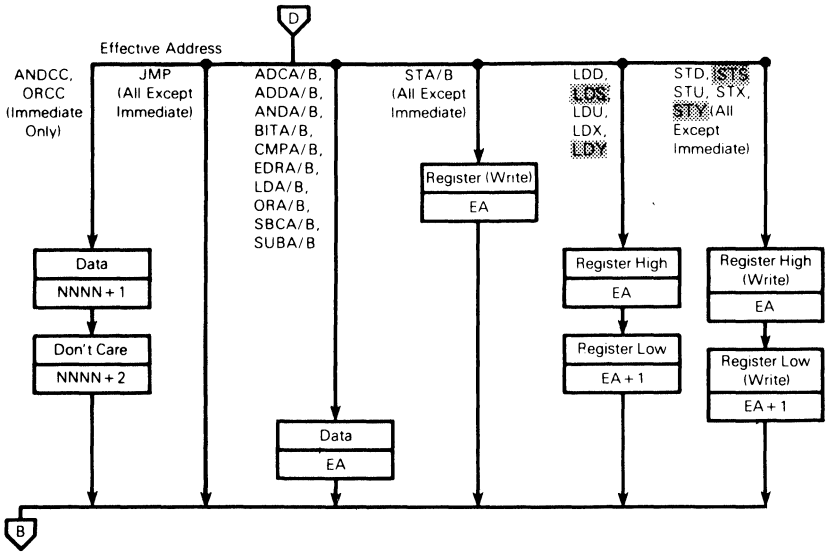
Constant Offset from PC
- 8-Bit Offset — Program Counter + Offset Byte
- 16-bit Offset — Program Counter + Offset High Byte: Offset Low Byte

Extended Indirect
- 16-Bit Address — Address High Byte: Addres Low Byte

XXXX

* The index register is incremented following the indexed access

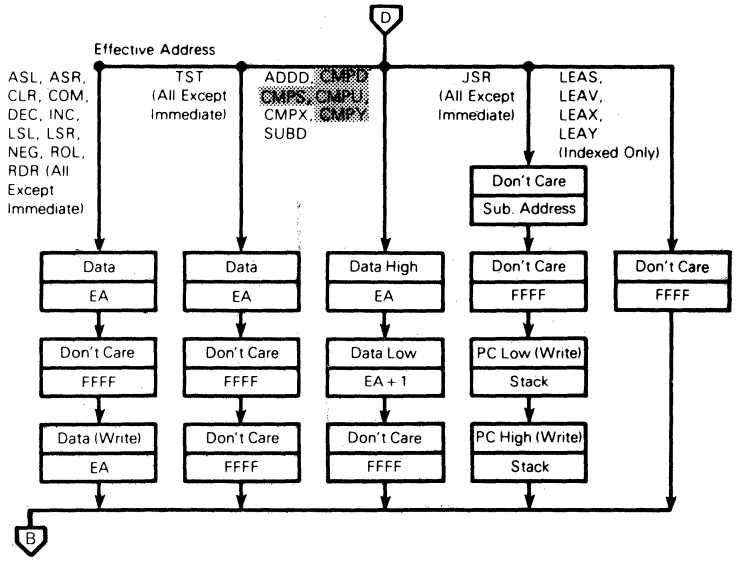
FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 8 of 9)



	Effective Address (EA)
Constant Offset from R	
No Offset	Index Register
5-Bit Offset	Index Register
8-Bit Offset	Index Register + Post Byte
16-Bit Offset	Index Register + Post Byte High: Post Byte Low
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 1	Index Register*
Increment by 2	Index Register*
Decrement by 1	Index Register – 1
Decrement by 2	Index Register – 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-Bit Offset	Program Counter + Offset High Byte, Offset Low Byte
Direct	Direct Page Register, Address Low
Extended	Address High, Address Low
Immediate	NNNN + 1

*The index register is incremented following the indexed access

FIGURE 18 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 9 of 9)



	Effective Address (EA)
Constant Offset from R	
No Offset	Index Register
5-Bit Offset	Index Register
8-Bit Offset	Index Register + Post Byte
16-Bit Offset	Index Register + Post Byte High: Post Byte Low
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 1	Index Register*
Increment by 2	Index Register*
Decrement by 1	Index Register – 1
Decrement by 2	Index Register – 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-Bit Offset	Program Counter + Offset High Byte: Offset Low Byte
Direct	Direct Page Register: Address Low
Extended	Address High: Address Low
Immediate	NNNN + 1

* The index register is incremented following the indexed access

TABLE 4 — 8-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADCA, ADCB	Add memory to accumulator with carry
ADDA, ADDB	Add memory to accumulator
ANDA, ANDB	And memory with accumulator
ASL, ASLA, ASLB	Arithmetic shift of accumulator or memory left
ASR, ASRA, ASRB	Arithmetic shift of accumulator or memory right
BITA, BITB	Bit test memory with accumulator
CLR, CLRA, CLRB	Clear accumulator or memory location
CMPA, CMPB	Compare memory from accumulator
COM, COMA, COMB	Complement accumulator or memory location
DAA	Decimal adjust A accumulator
DEC, DECA, DECB	Decrement accumulator or memory location
EORA, EORB	Exclusive or memory with accumulator
EXG R1, R2	Exchange R1 with R2 (R1, R2 = A, B, CC, DP)
INC, INCA, INCB	Increment accumulator or memory location
LDA, LDB	Load accumulator from memory
LSL, LSLA, LSLB	Logical shift left accumulator or memory location
LSR, LSRA, LSRB	Logical shift right accumulator or memory location
MUL	Unsigned multiply (A × B → D)
NEG, NEGA, NEGB	Negate accumulator or memory
ORA, ORB	Or memory with accumulator
ROL, ROLA, ROLB	Rotate accumulator or memory left
ROR, RORA, RORB	Rotate accumulator or memory right
SBCA, SBCB	Subtract memory from accumulator with borrow
STA, STB	Store accumulator to memory
SUBA, SUBB	Subtract memory from accumulator
TST, TSTA, TSTB	Test accumulator or memory location
TFR R1, R2	Transfer R1 to R2 (R1, R2 = A, B, CC, DP)

NOTE: A, B, CC, or DP may be pushed to (pulled from) stack with either PSHS, PSHU (PULS, PULU) instructions.

TABLE 5 — 16-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADDD	Add memory to D accumulator
CMPD	Compare memory from D accumulator
EXG D, R	Exchange D with X, Y, S, U, or PC
LDD	Load D accumulator from memory
SEX	Sign Extend B accumulator into A accumulator
STD	Store D accumulator to memory
SUBD	Subtract memory from D accumulator
TFR D, R	Transfer D to X, Y, S, U, or PC
TFR R, D	Transfer X, Y, S, U, or PC to D

NOTE: D may be pushed (pulled) to stack with either PSHS, PSHU (PULS, PULU) instructions.

TABLE 6 — INDEX REGISTER/STACK POINTER INSTRUCTIONS

Instruction	Description
CMPS, CMPU	Compare memory from stack pointer
CMPX, CMPY	Compare memory from index register
EXG R1, R2	Exchange D, X, Y, X, U, or PC with D, X Y, S, U, or PC
LEAS, LEAU	Load effective address into stack pointer
LEAX, LEAY	Load effective address into index register
LDS, LDU	Load stack pointer from memory
LDX, LDY	Load index register from memory
PSHS	Push A, B, CC, DP, D, X, Y, U, or PC onto hardware stack
PSHU	Push A, B, CC, DP, D, X, Y, S, or PC onto user stack
PULS	Pull A, B, CC, DP, D, X, Y, U, or PC from hardware stack
PULU	Pull A, B, CC, DP, D, X, Y, S, or PC from hardware stack
STS, STU	Store stack pointer to memory
STX, STY	Store index register to memory
TFR R1, R2	Transfer D, X, Y, S, U or PC to D, X, Y, S, U, or PC
ABX	Add B accumulator to X (unsigned)

TABLE 7 — BRANCH INSTRUCTIONS

Instruction	Description
SIMPLE BRANCHES	
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BMI, LBMI	Branch if minus
BPL, LBPL	Branch if plus
BCS, LBCS	Branch if carry set
BCC, LBCC	Branch if carry clear
BVS, LBVS	Branch if overflow set
BVC, LBVC	Branch if overflow clear
SIGNED BRANCHES	
BGT, LBGT	Branch if greater (signed)
BVS, LBVS	Branch if invalid 2s complement result
BGE, LBGE	Branch if greater than or equal (signed)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLE, LBLE	Branch if less than or equal (signed)
BVC, LBVC	Branch if valid 2s complement result
BLT, LBLT	Branch if less than (signed)
UNSIGNED BRANCHES	
BHI, LBHI	Branch if higher (unsigned)
BCC, LBCC	Branch if higher or same (unsigned)
BHS, LBHS	Branch if higher or same (unsigned)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLS, LBLS	Branch if lower or same (unsigned)
BCS, LBCS	Branch if lower (unsigned)
BLO, LBLO	Branch if lower (unsigned)
OTHER BRANCHES	
BSR, LBSR	Branch to subroutine
BRA, LBRA	Branch always
BRN, LBRN	Branch never

TABLE 8 — MISCELLANEOUS INSTRUCTIONS

Instruction	Description
ANDCC	AND condition code register
CWAI	AND condition code register, then wait for interrupt
NOP	No operation
ORCC	OR condition code register
JMP	Jump
JSR	Jump to subroutine
RTI	Return from interrupt
RTS	Return from subroutine
SWI, SWI2, SWI3	Software interrupt (absolute indirect)
SYNC	Synchronize with interrupt line

TABLE 9 – HEXADECIMAL VALUES OF MACHINE CODES

OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#
00	NEG	Direct	6	2	30	LEAX	Indexed	4+	2+	60	NEG	Indexed	6+	2+
01	*				31	LEAY		4+	2+	61	*			
02	*				32	LEAS		4+	2+	62	*			
03	COM		6	2	33	LEAU	Indexed	4+	2+	63	COM		6+	2+
04	LSR		6	2	34	PSHS	Immed	5+	2	64	LSR		6+	2+
05	*				35	PULS	Immed	5+	2	65	*			
06	ROR		6	2	36	PSHU	Immed	5+	2	66	ROR		6+	2+
07	ASR		6	2	37	PULU	Immed	5+	2	67	ASR		6+	2+
08	ASL, LSL		6	2	38	*	--			68	ASL, LSL		6+	2+
09	ROL		6	2	39	RTS	Inherent	5	1	69	ROL		6+	2+
0A	DEC		6	2	3A	ABX		3	1	6A	DEC		6+	2+
0B	*				3B	RTI		6/15	1	6B	*			
0C	INC		6	2	3C	CWAI		≥ 20	2	6C	INC		6+	2+
0D	TST		6	2	3D	MUL	Inherent	11	1	6D	TST		6+	2+
0E	JMP		3	2	3E	*	--			6E	JMP		3+	2+
0F	CLR	Direct	6	2	3F	SWI	Inherent	19	1	6F	CLR	Indexed	6+	2+
10	Page 2	–	–	–	40	NEGA	Inherent	2	1	70	NEG	Extended	7	3
11	Page 3	–	–	–	41	*				71	*			
12	NOP	Inherent	2	1	42	*				72	*			
13	SYNC	Inherent	≥ 4	1	43	COMA		2	1	73	COM		7	3
14	*				44	LSRA		2	1	74	LSR		7	3
15	*				45	*				75	*			
16	LBRA	Relative	5	3	46	RORA		2	1	76	ROR		7	3
17	LBSR	Relative	9	3	47	ASRA		2	1	77	ASR		7	3
18	*				48	ASLA, LSLA		2	1	78	ASL, LSL		7	3
19	DAA	Inherent	2	1	49	ROLA		2	1	79	ROL		7	3
1A	ORCC	Immed	3	2	4A	DECA		2	1	7A	DEC		7	3
1B	*	–			4B	*				7B	*			
1C	ANDCC	Immed	3	2	4C	INCA		2	1	7C	INC		7	3
1D	SEX	Inherent	2	1	4D	TSTA		2	1	7D	TST		7	3
1E	EXG	Immed	8	2	4E	*				7E	JMP		4	3
1F	TFR	Immed	6	2	4F	CLRA	Inherent	2	1	7F	CLR	Extended	7	3
20	BRA	Relative	3	2	50	NEGB	Inherent	2	1	80	SUBA	Immed	2	2
21	BRN		3	2	51	*				81	CMPA		2	2
22	BHI		3	2	52	*				82	SBCA		2	2
23	BLS		3	2	53	COMB		2	1	83	SUBD		4	3
24	BHS, BCC		3	2	54	LSRB		2	1	84	ANDA		2	2
25	BLO, BCS		3	2	55	*				85	BITA		2	2
26	BNE		3	2	56	RORB		2	1	86	LDA		2	2
27	BEQ		3	2	57	ASRB		2	1	87	*			
28	BVC		3	2	58	ASLB, LSLB		2	1	88	EORA		2	2
29	BVS		3	2	59	ROLB		2	1	89	ADCA		2	2
2A	BPL		3	2	5A	DECB		2	1	8A	ORA		2	2
2B	BMI		3	2	5B	*				8B	ADDA		2	2
2C	BGE		3	2	5C	INCB		2	1	8C	CMPX	Immed	4	3
2D	BLT		3	2	5D	TSTB		2	1	8D	BSR	Relative	7	2
2E	BGT		3	2	5E	*				8E	LDX	Immed	3	3
2F	BLE	Relative	3	2	5F	CLRB	Inherent	2	1	8F	*			

LEGEND:

~ Number of MPU cycles (less possible push pull or indexed-mode cycles)

Number of program bytes

* Denotes unused opcode

TABLE 9 – HEXADECIMAL VALUES OF MACHINE CODES (CONTINUED)

OP	Mnem	Mode	~	#
90	SUBA	Direct	4	2
91	CMPA		4	2
92	SBCA		4	2
93	SUBD		6	2
94	ANDA		4	2
95	BITA		4	2
96	LDA		4	2
97	STA		4	2
98	EORA		4	2
99	ADCA		4	2
9A	ORA		4	2
9B	ADDA		4	2
9C	CMPX		6	2
9D	JSR		7	2
9E	LDX		5	2
9F	STX	Direct	5	2
A0	SUBA	Indexed	4+	2+
A1	CMPA		4+	2+
A2	SBCA		4+	2+
A3	SUBD		6+	2+
A4	ANDA		4+	2+
A5	BITA		4+	2+
A6	LDA		4+	2+
A7	STA		4+	2+
A8	EORA		4+	2+
A9	ADCA		4+	2+
AA	ORA		4+	2+
AB	ADDA		4+	2+
AC	CMPX		6+	2+
AD	JSR		7+	2+
AE	LDX		5+	2+
AF	STX	Indexed	5+	2+
B0	SUBA	Extended	5	3
B1	CMPA		5	3
B2	SBCA		5	3
B3	SUBD		7	3
B4	ANDA		5	3
B5	BITA		5	3
B6	LDA		5	3
B7	STA		5	3
B8	EORA		5	3
B9	ADCA		5	3
BA	ORA		5	3
BB	ADDA		5	3
BC	CMPX		7	3
BD	JSR		8	3
BE	LDX		6	3
BF	STX	Extended	6	3

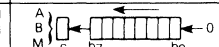
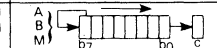
NOTE: All unused opcodes are both undefined and illegal

OP	Mnem	Mode	~	#
C0	SUBB	Immed	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB	Immed	2	2
C6	LDB	Immed	2	2
C7	*			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	*			
CE	LDU	Immed	3	3
CF	*			
D0	SUBB	Direct	4	2
D1	CMPB		4	2
D2	SBCB		4	2
D3	ADDD		6	2
D4	ANDB		4	2
D5	BITB		4	2
D6	LDB		4	2
D7	STB		4	2
D8	EORB		4	2
D9	ADCB		4	2
DA	ORB		4	2
DB	ADDB		4	2
DC	LDD		5	2
DD	STD		5	2
DE	LDU		5	2
DF	STU	Direct	5	2
E0	SUBB	Indexed	4+	2+
E1	CMPB		4+	2+
E2	SBCB		4+	2+
E3	ADDD		6+	2+
E4	ANDB		4+	2+
E5	BITB		4+	2+
E6	LDB		4+	2+
E7	STB		4+	2+
E8	EORB		4+	2+
E9	ADCB		4+	2+
EA	ORB		4+	2+
EB	ADDB		4+	2+
EC	LDD		5+	2+
ED	STD		5+	2+
EE	LDU		5+	2+
EF	STU	Indexed	5+	2+
F0	SUBB	Extended	5	3
F1	CMPB		5	3
F2	SBCB		5	3
F3	ADDD		7	3
F4	ANDB		5	3
F5	BITB		5	3
F6	LDB		5	3
F7	STB		5	3
F8	EORB		5	3
F9	ADCB		5	3
FA	ORB		5	3
FB	ADDB	Extended	5	3
FC	LDD	Extended	6	3
FD	STD		6	3
FE	LDU		6	3
FF	STU	Extended	6	3

Page 2 and 3 Machine Codes

OP	Mnem	Mode	~	#
1021	LBRN	Relative	5	4
1022	LBHI		5(6)	4
1023	LBLS		5(6)	4
1024	LBHS, LBCC		5(6)	4
1025	LBCS, LBLO		5(6)	4
1026	LBNE		5(6)	4
1027	LBEQ		5(6)	4
1028	LBVC		5(6)	4
1029	LBVS		5(6)	4
102A	LBPL		5(6)	4
102B	LBMI		5(6)	4
102C	LBGE		5(6)	4
102D	LBLT		5(6)	4
102E	LBGT		5(6)	4
102F	LBLE	Relative	5(6)	4
103F	SWI2	Inherent	20	2
1083	CMPD	Immed	5	4
108C	CMPY		5	4
108E	LDY	Immed	4	4
1093	CMPD	Direct	7	3
109C	CMPY		7	3
109E	LDY		6	3
109F	STY	Direct	6	3
10A3	CMPD	Indexed	7+	3+
10AC	CMPY		7+	3+
10AE	LDY		6+	3+
10AF	STY	Indexed	6+	3+
10B3	CMPD	Extended	8	4
10BC	CMPY		8	4
10BE	LDY		7	4
10BF	STY	Extended	7	4
10CE	LDS	Immed	4	4
10DE	LDS	Direct	6	3
10DF	STS	Direct	6	3
10EE	LDS	Indexed	6+	3+
10EF	STS	Indexed	6+	3+
10FE	LDS	Extended	7	4
10FF	STS	Extended	7	4
113F	SWI3	Inherent	20	2
1183	CMPU	Immed	5	4
118C	CMPS	Immed	5	4
1193	CMPU	Direct	7	3
119C	CMPS	Direct	7	3
11A3	CMPU	Indexed	7+	3+
11AC	CMPS	Indexed	7+	3+
11B3	CMPU	Extended	8	4
11BC	CMPS	Extended	8	4

FIGURE 19 — PROGRAMMING AID

Instruction	Forms	Addressing Modes: Immediate Op	Immediate ~	Immediate #	Direct Op	Direct ~	Direct #	Indexed Op	Indexed ~	Indexed #	Extended Op	Extended ~	Extended #	Inherent Op	Inherent ~	Inherent #	Description	5 H	3 N	2 Z	1 V	0 C
ABX														3A	3	1	B + X → X (Unsigned)	•	•	•	•	•
ADC	ADCA	89	2	2	99	4	2	A9	4 +	2 +	B9	5	3				A + M + C → A	↕	↕	↕	↕	↕
	ADCB	C9	2	2	D9	4	2	E9	4 +	2 +	F9	5	3				B + M + C → B	↕	↕	↕	↕	↕
ADD	ADDA	8B	2	2	9B	4	2	AB	4 +	2 +	BB	5	3				A + M → A	↕	↕	↕	↕	↕
	ADDB	CB	2	2	DB	4	2	EB	4 +	2 +	FB	5	3				B + M → B	↕	↕	↕	↕	↕
	ADDD	C3	4	3	D3	6	2	E3	6 +	2 +	F3	7	3				D + M:M + 1 → D	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	4	2	A4	4 +	2 +	B4	5	3				A ∧ M → A	•	↕	↕	0	•
	ANDB	C4	2	2	D4	4	2	E4	4 +	2 +	F4	5	3				B ∧ M → B	•	↕	↕	0	•
	ANDCC	1C	3	2													CC ∧ IMM → CC					?
ASL	ASLA													48	2	1		8	↕	↕	↕	↕
	ASLB													58	2	1		8	↕	↕	↕	↕
	ASL				08	6	2	68	6 +	2 +	78	7	3					8	↕	↕	↕	↕
ASR	ASRA													47	2	1		8	↕	↕	•	↕
	ASRB													57	2	1		8	↕	↕	•	↕
	ASR				07	6	2	67	6 +	2 +	77	7	3					8	↕	↕	•	↕
BIT	BITA	85	2	2	95	4	2	A5	4 +	2 +	B5	5	3				Bit Test A (M ∧ A)	•	↕	↕	0	•
	BITB	C5	2	2	D5	4	2	E5	4 +	2 +	F5	5	3				Bit Test B (M ∧ B)	•	↕	↕	0	•
CLR	CLRA													4F	2	1	0 → A	•	0	1	0	0
	CLRB													5F	2	1	0 → B	•	0	1	0	0
	CLR				0F	6	2	6F	6 +	2 +	7F	7	3				0 → M	•	0	1	0	0
CMP	CMPA	81	2	2	91	4	2	A1	4 +	2 +	B1	5	3				Compare M from A	↕	↕	↕	↕	↕
	CMPB	C1	2	2	D1	4	2	E1	4 +	2 +	F1	5	3				Compare M from B	↕	↕	↕	↕	↕
	CMPD	10	5	4	10	7	3	10	7 +	3 +	10	8	4				Compare M:M + 1 from D	•	↕	↕	↕	↕
		83			93			A3			B3											
	CMPS	11	5	4	11	7	3	11	7 +	3 +	11	8	4				Compare M:M + 1 from S	•	↕	↕	↕	↕
		8C			9C			AC			BC											
	CMPU	11	5	4	11	7	3	11	7 +	3 +	11	8	4				Compare M:M + 1 from U	•	↕	↕	↕	↕
		83			93			A3			B3											
	CMPX	8C	4	3	9C	6	2	AC	6 +	2 +	BC	7	3				Compare M:M + 1 from X	•	↕	↕	↕	↕
	CMPY	10	5	4	10	7	3	10	7 +	3 +	10	8	4				Compare M:M + 1 from Y	•	↕	↕	↕	↕
		8C			9C			AC			BC											
COM	COMA													43	2	1	$\overline{A} \rightarrow A$	•	↕	↕	0	1
	COMB													53	2	1	$\overline{B} \rightarrow B$	•	↕	↕	0	1
	COM				03	6	2	63	6 +	2 +	73	7	3				$\overline{M} \rightarrow M$	•	↕	↕	0	1
CWAI		3C	≥20	2													CC ∧ IMM → CC Wait for Interrupt					?
DAA														19	2	1	Decimal Adjust A	•	↕	↕	0	↕
DEC	DECA													4A	2	1	A − 1 → A	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	↕	↕	↕	•
	DEC				0A	6	2	6A	6 +	2 +	7A	7	3				M − 1 → M	•	↕	↕	↕	•
EOR	EORA	88	2	2	98	4	2	A8	4 +	2 +	B8	5	3				A ⊻ M → A	•	↕	↕	0	•
	EORB	C8	2	2	D8	4	2	E8	4 +	2 +	F8	5	3				B ⊻ M → B	•	↕	↕	0	•
EXG	R1, R2	1E	8	2													R1 ↔ R2[2]	•	•	•	•	•
INC	INCA													4C	2	1	A + 1 → A	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	↕	↕	↕	•
	INC				0C	6	2	6C	6 +	2 +	7C	7	3				M + 1 → M	•	↕	↕	↕	•
JMP					0E	3	2	6E	3 +	2 +	7E	4	3				EA[3] → PC	•	•	•	•	•
JSR					9D	7	2	AD	7 +	2 +	BD	8	3				Jump to Subroutine	•	•	•	•	•
LD	LDA	86	2	2	96	4	2	A6	4 +	2 +	B6	5	3				M → A	•	↕	↕	0	•
	LDB	C6	2	2	D6	4	2	E6	4 +	2 +	F6	5	3				M → B	•	↕	↕	0	•
	LDD	CC	3	3	DC	5	2	EC	5 +	2 +	FC	6	3				M:M + 1 → D	•	↕	↕	0	•
	LDS	10	4	4	10	6	3	10	6 +	3 +	10	7	4				M:M + 1 → S	•	↕	↕	0	•
		CE			DE			EE			FE											
	LDU	CE	3	3	DE	5	2	EE	5 +	2 +	FE	6	3				M:M + 1 → U	•	↕	↕	0	•
	LDX	8E	3	3	9E	5	2	AE	5 +	2 +	BE	6	3				M:M + 1 → X	•	↕	↕	0	•
	LDY	10	4	4	10	6	3	10	6 +	3 +	10	7	4				M:M + 1 → Y	•	↕	↕	0	•
		8E			9E			AE			BE											
LEA	LEAS							32	4 +	2 +							EA[3] → S	•	•	•	•	•
	LEAU							33	4 +	2 +							EA[3] → U	•	•	•	•	•
	LEAX							30	4 +	2 +							EA[3] → X	•	•	↕	•	•
	LEAY							31	4 +	2 +							EA[3] → Y	•	•	↕	•	•

LEGEND:

OP	Operation Code (Hexadecimal)	$\overline{M}$	Complement of M	↕	Test and set if true, cleared otherwise
~	Number of MPU Cycles	→	Transfer Into	•	Not Affected
#	Number of Program Bytes	H	Half-carry (from bit 3)	CC	Condition Code Register
+	Arithmetic Plus	N	Negative (sign bit)	:	Concatenation
−	Arithmetic Minus	Z	Zero result	V	Logical or
•	Multiply	V	Overflow, 2's complement	∧	Logical and
		C	Carry from ALU	⊻	Logical Exclusive or

FIGURE 19 — PROGRAMMING AID (CONTINUED)

Instruction	Forms	Addressing Modes: Immediate Op	Immediate ~	Immediate #	Direct Op	Direct ~	Direct #	Indexed[1] Op	Indexed ~	Indexed #	Extended Op	Extended ~	Extended #	Inherent Op	Inherent ~	Inherent #	Description	5 H	3 N	2 Z	1 V	0 C
LSL	LSLA													48	2	1	A, B, M: c b7 … b0 ← 0	•	↕	↕	↕	↕
	LSLB													58	2	1		•	↕	↕	↕	↕
	LSL				08	6	2	68	6+	2+	78	7	3					•	↕	↕	↕	↕
LSR	LSRA													44	2	1	A, B, M: 0 → b7 … b0 → c	•	0	↕	•	↕
	LSRB													54	2	1		•	0	↕	•	↕
	LSR				04	6	2	64	6+	2+	74	7	3					•	0	↕	•	↕
MUL														3D	11	1	A × B → D (Unsigned)	•	•	↕	•	9
NEG	NEGA													40	2	1	$\overline{A}$ + 1 → A	8	↕	↕	↕	↕
	NEGB													50	2	1	$\overline{B}$ + 1 → B	8	↕	↕	↕	↕
	NEG				00	6	2	60	6+	2+	70	7	3				$\overline{M}$ + 1 → M	8	↕	↕	↕	↕
NOP														12	2	1	No Operation	•	•	•	•	•
OR	ORA	8A	2	2	9A	4	2	AA	4+	2+	BA	5	3				A V M → A	•	↕	↕	0	•
	ORB	CA	2	2	DA	4	2	EA	4+	2+	FA	5	3				B V M → B	•	↕	↕	0	•
	ORCC	1A	3	2													CC V IMM → CC				7	
PSH	PSHS	34	5+4	2													Push Registers on S Stack	•	•	•	•	•
	PSHU	36	5+4	2													Push Registers on U Stack	•	•	•	•	•
PUL	PULS	35	5+4	2													Pull Registers from S Stack	•	•	•	•	•
	PULU	37	5+4	2													Pull Registers from U Stack	•	•	•	•	•
ROL	ROLA													49	2	1	A, B, M: c b7 … b0	•	↕	↕	↕	↕
	ROLB													59	2	1		•	↕	↕	↕	↕
	ROL				09	6	2	69	6+	2+	79	7	3					•	↕	↕	↕	↕
ROR	RORA													46	2	1	A, B, M: c b7 … b0	•	↕	↕	•	↕
	RORB													56	2	1		•	↕	↕	•	↕
	ROR				06	6	2	66	6+	2+	76	7	3					•	↕	↕	•	↕
RTI														3B	6/15	1	Return From Interrupt					7
RTS														39	5	1	Return from Subroutine	•	•	•	•	•
SBC	SBCA	82	2	2	92	4	2	A2	4+	2+	B2	5	3				A - M - C → A	8	↕	↕	↕	↕
	SBCB	C2	2	2	D2	4	2	E2	4+	2+	F2	5	3				B - M - C → B	8	↕	↕	↕	↕
SEX														1D	2	1	Sign Extend B into A	•	↕	↕	0	•
ST	STA				97	4	2	A7	4+	2+	B7	5	3				A → M	•	↕	↕	0	•
	STB				D7	4	2	E7	4+	2+	F7	5	3				B → M	•	↕	↕	0	•
	STD				DD	5	2	ED	5+	2+	FD	6	3				D → M:M + 1	•	↕	↕	0	•
	STS				10	6	3	10	6+	3+	10	7	4				S → M:M + 1	•	↕	↕	0	•
					DF			EF			FF											
	STU				DF	5	2	EF	5+	2+	FF	6	3				U → M:M + 1	•	↕	↕	0	•
	STX				9F	5	2	AF	5+	2+	BF	6	3				X → M:M + 1	•	↕	↕	0	•
	STY				10	6	3	10			10	7	4				Y → M:M + 1	•	↕	↕	0	•
					9F			AF	6+	3+	BF											
SUB	SUBA	80	2	2	90	4	2	A0	4+	2+	B0	5	3				A - M → A	8	↕	↕	↕	↕
	SUBB	C0	2	2	D0	4	2	E0	4+	2+	F0	5	3				B - M → B	8	↕	↕	↕	↕
	SUBD	83	4	3	93	6	2	A3	6+	2+	B3	7	3				D - M:M + 1 → D	•	↕	↕	↕	↕
SWI	SWI[6]													3F	19	1	Software Interrupt 1	•	•	•	•	•
	SWI2[6]													10 3F	20	2	Software Interrupt 2	•	•	•	•	•
	SWI3[6]													11 3F	20	1	Software Interrupt 3	•	•	•	•	•
SYNC														13	≥4	1	Synchronize to Interrupt	•	•	•	•	•
TFR	R1, R2	1F	6	2													R1 → R2[2]	•	•	•	•	•
TST	TSTA													4D	2	1	Test A	•	↕	↕	0	•
	TSTB													5D	2	1	Test B	•	↕	↕	0	•
	TST				0D	6	2	6D	6+	2+	7D	7	3				Test M	•	↕	↕	0	•

NOTES:

1. This column gives a base cycle and byte count. To obtain total count, add the values obtained from the INDEXED ADDRESSING MODE table, Table 2.
2. R1 and R2 may be any pair of 8 bit or any pair of 16 bit registers.
   The 8 bit registers are: A, B, CC, DP
   The 16 bit registers are: X, Y, U, S, D, PC
3. EA is the effective address.
4. The PSH and PUL instructions require 5 cycles plus 1 cycle for each **byte** pushed or pulled.
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken (Branch instructions).
6. SWI sets I and F bits. SWI2 and SWI3 do not affect I and F.
7. Conditions Codes set as a direct result of the instruction.
8. Vaue of half-carry flag is undefined.
9. Special Case — Carry set if b7 is SET.

FIGURE 19 — PROGRAMMING AID (CONTINUED)

Branch Instructions

Instruction	Forms	Addressing Mode Relative OP	~ 5	#	Description	5 H	3 N	2 Z	1 V	0 C
BCC	BCC	24	3	2	Branch C = 0	•	•	•	•	•
	LBCC	10	5(6)	4	Long Branch	•	•	•	•	•
		24			C = 0					
BCS	BCS	25	3	2	Branch C = 1	•	•	•	•	•
	LBCS	10	5(6)	4	Long Branch	•	•	•	•	•
		25			C = 1					
BEQ	BEQ	27	3	2	Branch Z = 1	•	•	•	•	•
	LBEQ	10	5(6)	4	Long Branch	•	•	•	•	•
		27			Z = 0					
BGE	BGE	2C	3	2	Branch ≥ Zero	•	•	•	•	•
	LBGE	10	5(6)	4	Long Branch ≥ Zero	•	•	•	•	•
		2C								
BGT	BGT	2E	3	2	Branch > Zero	•	•	•	•	•
	LBGT	10	5(6)	4	Long Branch > Zero	•	•	•	•	•
		2E								
BHI	BHI	22	3	2	Branch Higher	•	•	•	•	•
	LBHI	10	5(6)	4	Long Branch Higher	•	•	•	•	•
		22								
BHS	BHS	24	3	2	Branch Higher or Same	•	•	•	•	•
	LBHS	10	5(6)	4	Long Branch Higher or Same	•	•	•	•	•
		24								
BLE	BLE	2F	3	2	Branch ≤ Zero	•	•	•	•	•
	LBLE	10	5(6)	4	Long Branch ≤ Zero	•	•	•	•	•
		2F								
BLO	BLO	25	3	2	Branch lower	•	•	•	•	•
	LBLO	10	5(6)	4	Long Branch Lower	•	•	•	•	•
		25								

Instruction	Forms	Addressing Mode Relative OP	~ 5	#	Description	5 H	3 N	2 Z	1 V	0 C
BLS	BLS	23	3	2	Branch Lower or Same	•	•	•	•	•
	LBLS	10	5(6)	4	Long Branch Lower or Same	•	•	•	•	•
		23								
BLT	BLT	2D	3	2	Branch < Zero	•	•	•	•	•
	LBLT	10	5(6)	4	Long Branch < Zero	•	•	•	•	•
		2D								
BMI	BMI	2B	3	2	Branch Minus	•	•	•	•	•
	LBMI	10	5(6)	4	Long Branch Minus	•	•	•	•	•
		2B								
BNE	BNE	26	3	2	Branch Z = 0	•	•	•	•	•
	LBNE	10	5(6)	4	Long Branch Z ≠ 0	•	•	•	•	•
		26								
BPL	BPL	2A	3	2	Branch Plus	•	•	•	•	•
	LBPL	10	5(6)	4	Long Branch Plus	•	•	•	•	•
		2A								
BRA	BRA	20	3	2	Branch Always	•	•	•	•	•
	LBRA	16	5	3	Long Branch Always	•	•	•	•	•
BRN	BRN	21	3	2	Branch Never	•	•	•	•	•
	LBRN	10	5	4	Long Branch Never	•	•	•	•	•
		21								
BSR	BSR	8D	7	2	Branch to Subroutine	•	•	•	•	•
	LBSR	17	9	3	Long Branch to Subroutine	•	•	•	•	•
BVC	BVC	28	3	2	Branch V = 0	•	•	•	•	•
	LBVC	10	5(6)	4	Long Branch V = 0	•	•	•	•	•
		28								
BVS	BVS	29	3	2	Branch V = 1	•	•	•	•	•
	LBVS	10	5(6)	4	Long Branch V = 1	•	•	•	•	•
		29								

SIMPLE BRANCHES

	OP	~	#
BRA	20	3	2
LBRA	16	5	3
BRN	21	3	2
LBRN	1021	5	4
BSR	8D	7	2
LBSR	17	9	3

SIMPLE CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
N = 1	BMI	2B	BPL	2A
Z = 1	BEQ	27	BNE	26
V = 1	BVS	29	BVC	28
C = 1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BGT	2E	BLE	2F
r ≥ m	BGE	2C	BLT	2D
r = m	BEQ	27	BNE	26
r ≤ m	BLE	2F	BGT	2E
r < m	BLT	2D	BGE	2C

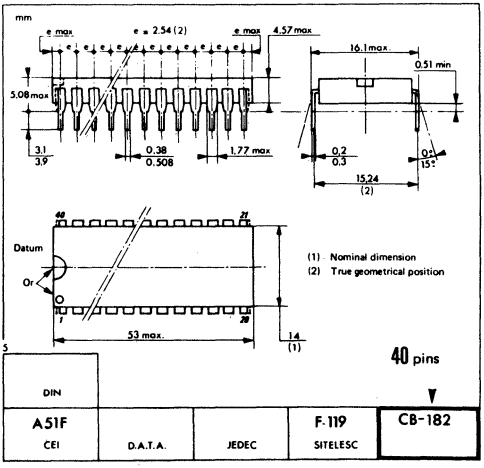
UNSIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BHI	22	BLS	23
r ≥ m	BHS	24	BLO	25
r = m	BEQ	27	BNE	26
r ≤ m	BLS	23	BHI	22
r < m	BLO	25	BHS	24

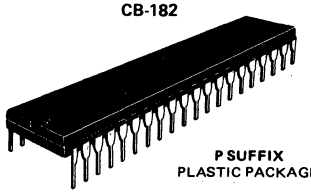
NOTES:

1. All conditional branches have both short and long variations.
2. All short branches are two bytes and require three cycles.
3. All conditional long branches are formed by prefixing the short branch opcode with $10 and using a 16-bit destination offset.
4. All conditional long branches require four bytes and six cycles if the branch is taken or five cycles if the branch is not taken.

PHYSICAL DIMENSIONS



CB-182



P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

J SUFFIX
CERDIP PACKAGE

C SUFFIX
CERAMIC PACKAGE

ORDERING INFORMATION

EF68A09 | C | M | B/B

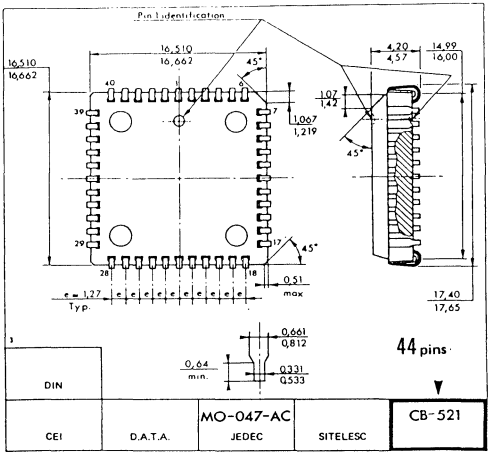
- EF68A09: Device
- C: Package
- M: Oper. temp.
- B/B: Screening level

The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

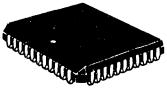
DEVICE	PACKAGE					OPER. TEMP			SCREENING LEVEL			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6809 (1.0 MHz)	●	●	●		●	●			●			
	●	●	●				●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68A09 (1.5 MHz)	●	●	●			●			●			
	●	●	●				●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68B09 (2.0 MHz)		●	●			●			●			
	●	●					●		●		●	

Examples : EF6809C, EF6809CV, EF6809CM

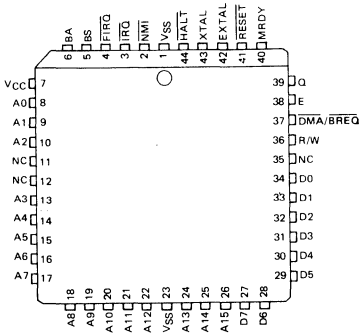
Package : **C** : Ceramic DIL, **J** : Cerdip DIL, **P** : Plastic DIL, **E** : LCCC, **FN** : PLCC.
Oper. temp. : **L*** : 0°C to +70°C, **V** : −40°C to +85°C, **M** : −55°C to +125°C, * : may be omitted.
Screening level : **Std** : (no-end suffix), **D** : NFC 96883 level D,
G/B : NFC 96883 level G, **B/B** : NFC 96883 level B and MIL-STD-883C level B.

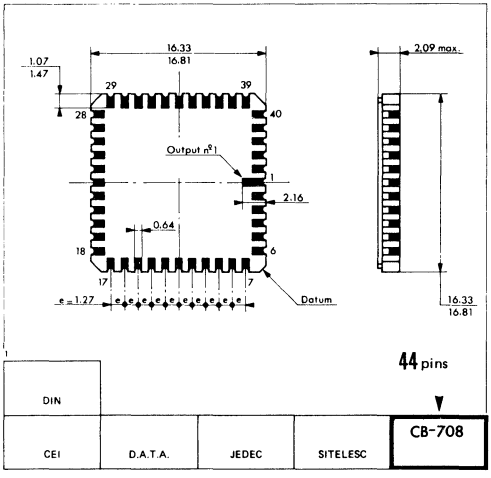


CB-521

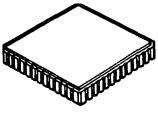


FN SUFFIX
PLCC 44





CB-708



E SUFFIX
LCCC 44

THOMSON COMPONENTS
MOSTEK

EF6809E

8-BIT MICROPROCESSOR UNIT (MPU)

The EF6809E is a revolutionary high performance 8-bit microprocessor which supports modern programming techniques such as position independence, reentrancy, and modular programming.

This third-generation addition to the 6800 Family has major architectural improvements which include additional registers, instructions, and addressing modes.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The EF6809E has the most complete set of addressing modes available on any 8-bit microprocessor today.

The EF6809E has hardware and software features which make it an ideal processor for higher level language execution or standard controller applications. External clock inputs are provided to allow synchronization with peripherals, systems, or other MPUs.

EF6800 COMPATIBLE

- Hardware – Interfaces with All 6800 Peripherals
- Software – Upward Source Code Compatible Instruction Set and Addressing Modes

ARCHITECTURAL FEATURES

- Two 16-Bit Index Registers
- Two 16-Bit Indexable Stack Pointers
- Two 8-Bit Accumulators can be Concatenated to Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

HARDWARE FEATURES

- External Clock Inputs, E and Q, Allow Synchronization
- TSC Input Controls Internal Bus Buffers
- LIC Indicates Opcode Fetch
- AVMA Allows Efficient Use of Common Resources in a Multiprocessor System
- BUSY is a Status Line for Multiprocessing
- Fast Interrupt Request Input Stacks Only Condition Code Register and Program Counter
- Interrupt Acknowledge Output Allows Vectoring By Devices
- Sync Acknowledge Output Allows for Synchronization to External Event
- Single Bus-Cycle $\overline{\text{RESET}}$
- Single 5-Volt Supply Operation
- $\overline{\text{NMI}}$ Inhibited After $\overline{\text{RESET}}$ Until After First Load of Stack Pointer
- Early Address Valid Allows Use With Slower Memories
- Early Write Data for Dynamic Memories

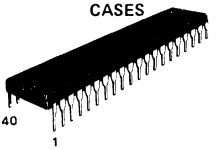
SOFTWARE FEATURES

- 10 Addressing Modes
  - 6800 Upward Compatible Addressing Modes
  - Direct Addressing Anywhere in Memory Map
  - Long Relative Branches
  - Program Counter Relative
  - True Indirect Addressing
  - Expanded Indexed Addressing
    0-, 5-, 8-, or 16-Bit Constant Offsets
    8- or 16-Bit Accumulator Offsets
    Auto-Increment/Decrement by 1 or 2
- Improved Stack Manipulation
- 1464 Instruction with Unique Addressing Modes
- 8 × 8 Unsigned Multiply
- 16-Bit Arithmetic
- Transfer/Exchange All Registers
- Push/Pull Any Registers or Any Set of Registers
- Load Effective Address

HMOS
(HIGH-DENSITY N-CHANNEL, SILICON-GATE)

8-BIT
MICROPROCESSING
UNIT

CASES

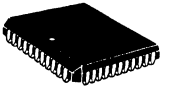


ALSO AVAILABLE

J SUFFIX
CERDIP PACKAGE

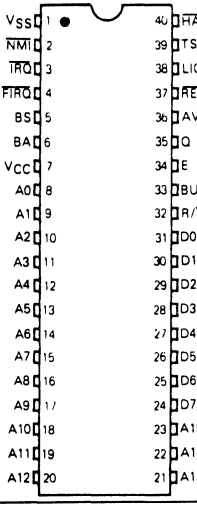
C SUFFIX
CERAMIC PACKAGE

CB-521



FN SUFFIX
PLCC 44

PIN ASSIGNMENT



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range EF6809E, EF68A09E, EF68B09E EF6809E, EF68A09E, EF68B09E, V suffix EF6809E, EF68A09E : M suffix	T_A	T_L to T_H 0 to +70 −40 to +85 −55 to +125	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit.

Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Cerdip Plastic PLCC	θ_{JA}	 50 60 100 100	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K + (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

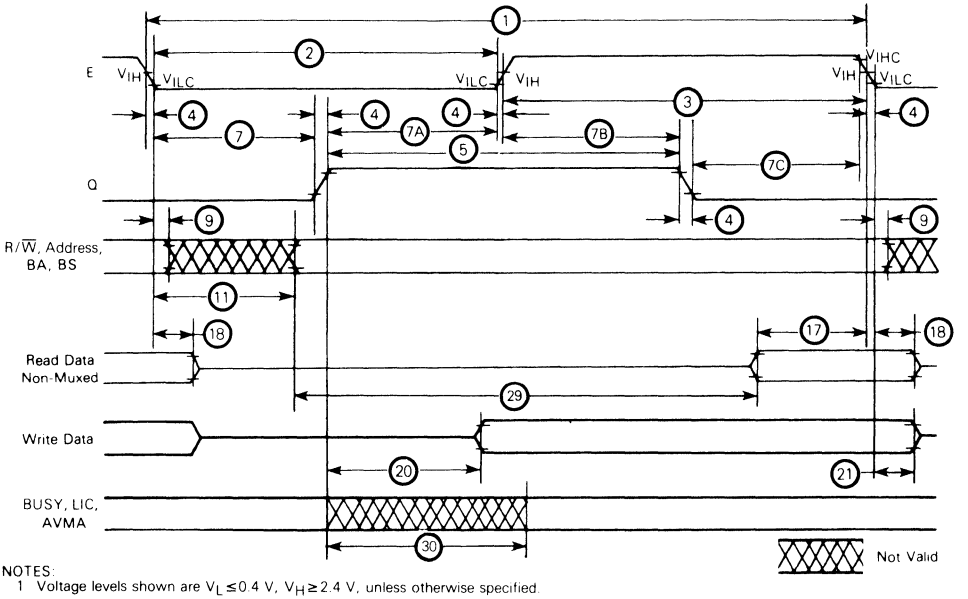
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, Q, $\overline{RESET}$	V_{IH}	V_{SS} + 2.0		V_{CC}	V
	$\overline{RESET}$	V_{IHR}	V_{SS} + 4.0	–	V_{CC}	
	E	V_{IHC}	V_{CC} − 0.75	–	V_{CC} + 0.3	
Input Low Voltage	Logic, $\overline{RESET}$	V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
	E	V_{ILC}	V_{SS} − 0.3	–	V_{SS} + 0.4	V
	Q	V_{ILQ}	V_{SS} − 0.3	–	V_{SS} + 0.6	V
Input Leakage Current (V_{in} = 0 to 5.25 V, V_{CC} = max)	Logic, Q, $\overline{RESET}$	I_{in}	–	–	2.5	μA
	E		–	–	100	
dc Output High Voltage (I_{Load} = −205 μA, V_{CC} = min) (I_{Load} = −145 μA, V_{CC} = min) (I_{Load} = −100 μA, V_{CC} = min)	D0-D7 A0-A15, R/$\overline{W}$ BA, BS, LIC, AVMA, BUSY	V_{OH}	V_{SS} + 2.4 V_{SS} + 2.4 V_{SS} + 2.4	– – –	– – –	V
dc Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = min)		V_{OL}	–	–	V_{SS} + 0.5	V
Internal Power Dissipation (Measured at T_A = 0°C in Steady State Operation)		P_{INT}	–	–	1.0	W
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7, Logic Inputs, Q, $\overline{RESET}$	C_{in}	–	10	15	pF
	E		–	30	50	
	A0-A15, R/$\overline{W}$, BA, BS, LIC, AVMA, BUSY	C_{out}	–	10	15	pF
Frequency of Operation (E and Q Inputs)	EF6809E	f	0.1	–	1.0	MHz
	EF68A09E		0.1	–	1.5	
	EF68B09E		0.1	–	2.0	
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = max)	D0-D7	I_{TSI}	–	2.0	10	μA
	A0-A15, R/$\overline{W}$		–	–	100	

*Capacitances are periodically tested rather than 100% tested.

BUS TIMING CHARACTERISTICS (See Notes 1, 2, 3, and 4)

Ident. Number	Characteristics	Symbol	EF6809E		EF68A09E		EF68B09E		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.667	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	450	9500	295	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	-	25	-	25	-	20	ns
5	Pulse Width, Q High	PW_{QH}	450	9500	280	9500	220	9500	ns
7	Delay Time, E to Q Rise	t_{EQ1}	200	-	130	-	100	-	ns
7A	Delay Time, Q High to E Rise	t_{EQ2}	200	-	130	-	100	-	ns
7B	Delay Time, E High to Q Fall	t_{EQ3}	200	-	130	-	100	-	ns
7C	Delay Time, Q High to E Fall	t_{EQ4}	200	-	130	-	100	-	ns
9	Address Hold Time	t_{AH}	20	-	20	-	20	-	ns
11	Address Delay Time from E Low (BA, BS, R/$\overline{W}$)	t_{AD}	-	200	-	140	-	110	ns
17	Read Data Setup Time	t_{DSR}	80	-	60	-	40	-	ns
18	Read Data Hold Time	t_{DHR}	10	-	10	-	10	-	ns
20	Data Delay Time from Q	t_{DDQ}	-	200	-	140	-	110	ns
21	Write Data Hold Time	t_{DHW}	30	-	30	-	30	-	ns
29	Usable Access Time	t_{ACC}	695	-	440	-	330	-	ns
30	Control Delay Time	t_{CD}	-	300	-	250	-	200	ns
	Interrupts, $\overline{HALT}$, $\overline{RESET}$, and TSC Setup Time (Figures 6, 7, 8, 9, 12, and 13)	t_{PCS}	200	-	140	-	110	-	ns
	TSC Drive to Valid Logic Level (Figure 13)	t_{TSV}	-	210	-	150	-	120	ns
	TSC Release MOS Buffers to High Impedance (Figure 13)	t_{TSR}	-	200	-	140	-	110	ns
	TSC Hi-Z Delay Time (Figure 13)	t_{TSD}	-	120	-	85	-	80	ns
	Processor Control Rise and Fall Time (Figure 7)	t_{PCr}, t_{PCf}	-	100	-	100	-	100	ns

FIGURE 1 — READ/WRITE DATA TO MEMORY OR PERIPHERALS TIMING DIAGRAM



NOTES:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Hold time (⑨) for BA and BS is not specified.
4. Usable access time is computed by: 1 – 4 – 11 max – 17

FIGURE 2 — EXPANDED BLOCK DIAGRAM

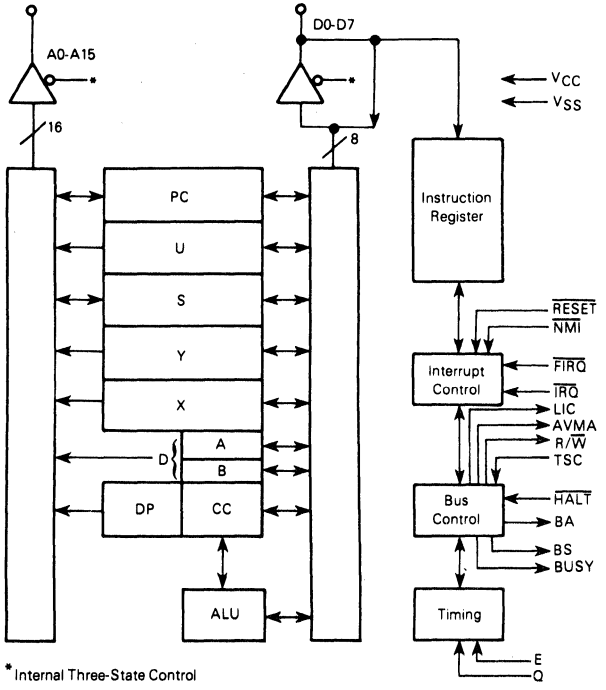
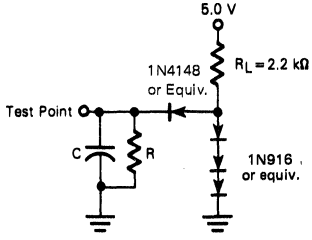


FIGURE 3 — BUS TIMING TEST LOAD



C = 30 pF for BA, BS, LIC, AVMA, BUSY
130 pF for D0-D7
90 pF for A0-A15, R/$\overline{W}$

R = 11.7 kΩ for D0-D7
16.5 kΩ for A0-A15, R/$\overline{W}$
24 kΩ for BA, BS, LIC, AVMA, BUSY

PROGRAMMING MODEL

As shown in Figure 4, the EF6809E adds three registers to the set available in the EF6800. The added registers include a direct page register, the user stack pointer, and a second index register.

ACCUMULATORS (A, B, D)

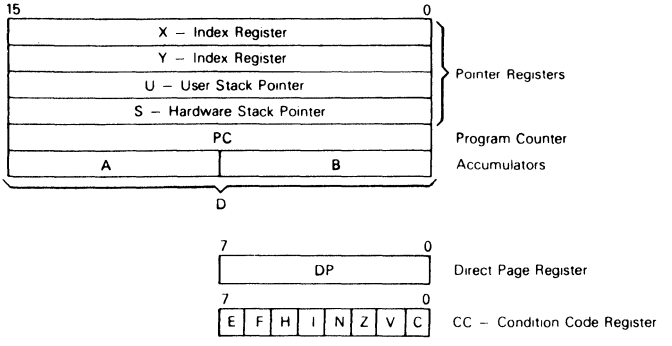
The A and B registers are general purpose accumulators which are used for arithmetic calculations and manipulation of data.

Certain instructions concatenate the A and B registers to form a single 16-bit accumulator. This is referred to as the D register, and is formed with the A register as the most significant byte.

DIRECT PAGE REGISTER (DP)

The direct page register of the EF6809E serves to enhance the direct addressing mode. The content of this register appears at the higher address outputs (A8-A15) during direct addressing instruction execution. This allows the direct mode to be used at any place in memory, under program control. To ensure 6800 compatibility, all bits of this register are cleared during processor reset.

FIGURE 4 PROGRAMMING MODEL OF THE MICROPROCESSING UNIT



INDEX REGISTERS (X, Y)

The index registers are used in indexed mode of addressing. The 16-bit address in this register takes part in the calculation of effective addresses. This address may be used to point to data directly or may be modified by an optional constant or register offset. During some indexed modes, the contents of the index register are incremented and decremented to point to the next item of tabular type data. All four pointer registers (X, Y, U, S) may be used as index registers.

STACK POINTER (U, S)

The hardware stack pointer (S) is used automatically by the processor during subroutine calls and interrupts. The user stack pointer (U) is controlled exclusively by the programmer. This allows arguments to be passed to and from subroutines with ease. The U register is frequently used as a stack marker. Both stack pointers have the same indexed mode addressing capabilities as the X and Y registers, but also support **Push** and **Pull** instructions. This allows the EF6809E to be used efficiently as a stack processor, greatly enhancing its ability to support higher level languages and modular programming.

NOTE

The stack pointers of the EF6809E point to the top of the stack in contrast to the EF6800 stack pointer, which pointed to the next free location on stack.

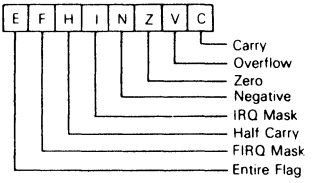
PROGRAM COUNTER

The program counter is used by the processor to point to the address of the next instruction to be executed by the processor. Relative addressing is provided allowing the program counter to be used like an index register in some situations.

CONDITION CODE REGISTER

The condition code register defines the state of the processor at any given time. See Figure 4.

FIGURE 5 – CONDITION CODE REGISTER FORMAT



CONDITION CODE REGISTER DESCRIPTION

BIT 0 (C)

Bit 0 is the carry flag and is usually the carry from the binary ALU. C is also used to represent a "borrow" from subtract like instructions (CMP, NEG, SUB, SBC) and is the complement of the carry from the binary ALU.

BIT 1 (V)

Bit 1 is the overflow flag and is set to a one by an operation which causes a signed twos complement arithmetic overflow. This overflow is detected in an operation in which the carry from the MSB in the ALU does not match the carry from the MSB-1.

BIT 2 (Z)

Bit 2 is the zero flag and is set to a one if the result of the previous operation was identically zero.

BIT 3 (N)

Bit 3 is the negative flag, which contains exactly the value of the MSB of the result of the preceding operation. Thus, a negative twos complement result will leave N set to a one.

BIT 4 (I)

Bit 4 is the $\overline{IRQ}$ mask bit. The processor will not recognize interrupts from the $\overline{IRQ}$ line if this bit is set to a one. $\overline{NMI}$, $\overline{FIRQ}$, $\overline{IRQ}$, $\overline{RESET}$, and SWI all set I to a one. SWI2 and SWI3 do not affect I.

BIT 5 (H)

Bit 5 is the half-carry bit, and is used to indicate a carry from bit 3 in the ALU as a result of an 8-bit addition only (ADC or ADD). This bit is used by the DAA instruction to perform a BCD decimal add adjust operation. The state of this flag is undefined in all subtract-like instructions.

BIT 6 (F)

Bit 6 is the $\overline{FIRQ}$ mask bit. The processor will not recognize interrupts from the $\overline{FIRQ}$ line if this bit is a one. $\overline{NMI}$, $\overline{FIRQ}$, SWI, and $\overline{RESET}$ all set F to a one. $\overline{IRQ}$, SWI2, and SWI3 do not affect F.

BIT 7 (E)

Bit 7 is the entire flag, and when set to a one indicates that the complete machine state (all the registers) was stacked, as opposed to the subset state (PC and CC). The E bit of the stacked CC is used on a return from interrupt (RTI) to determine the extent of the unstacking. Therefore, the current E left in the condition code register represents past action.

PIN DESCRIPTIONS

POWER (V_{SS}, V_{CC})

Two pins are used to supply power to the part: V_{SS} is ground or 0 volts, while V_{CC} is +5.0 V ±5%.

ADDRESS BUS (A0-A15)

Sixteen pins are used to output address information from the MPU onto the address bus. When the processor does not require the bus for a data transfer, it will output address $FFFF_{16}$, $R/\overline{W} = 1$, and BS = 0; this is a "dummy access" or $\overline{VMA}$ cycle. All address bus drivers are made high-impedance when output bus available (BA) is high or when TSC is asserted. Each pin will drive one Schottky TTL load or four LSTTL loads and 90 pF.

DATA BUS (D0-D7)

These eight pins provide communication with the system bidirectional data bus. Each pin will drive one Schottky TTL load or four LSTTL loads and 130 pF.

READ/WRITE ($R/\overline{W}$)

This signal indicates the direction of data transfer on the data bus. A low indicates that the MPU is writing data onto the data bus. $R/\overline{W}$ is made high impedance when BA is high or when TSC is asserted.

$\overline{RESET}$

A low level on this Schmitt-trigger input for greater than one bus cycle will reset the MPU, as shown in Figure 6. The reset vectors are fetched from locations $FFFE_{16}$ and $FFFF_{16}$ (Table 1) when interrupt acknowledge is true, ($\overline{BA} \bullet BS = 1$). During initial power on, the reset line should be held low until the clock input signals are fully operational.

Because the EF6809E $\overline{RESET}$ pin has a Schmitt-trigger input with a threshold voltage higher than that of standard peripherals, a simple R/C network may be used to reset the entire system. This higher threshold voltage ensures that all peripherals are out of the reset state before the processor.

$\overline{HALT}$

A low level on this input pin will cause the MPU to stop running at the end of the present instruction and remain halted indefinitely without loss of data. When halted, the BA output is driven high indicating the buses are high impedance. BS is also high which indicates the processor is in the halt state. While halted, the MPU will not respond to external real-time requests ($\overline{FIRQ}$, $\overline{IRQ}$) although $\overline{NMI}$ or $\overline{RESET}$ will be latched for later response. During the halt state, Q and E should continue to run normally. A halted state ($BA \bullet BS = 1$) can be achieved by pulling $\overline{HALT}$ low while $\overline{RESET}$ is still low. See Figure 7.

BUS AVAILABLE, BUS STATUS (BA, BS)

The bus available output is an indication of an internal control signal which makes the MOS buses of the MPU high impedance. When BA goes low, a dead cycle will elapse before the MPU acquires the bus. BA will not be asserted when TSC is active, thus allowing dead cycle consistency.

The bus status output signal, when decoded with BA, represents the MPU state (valid with leading edge of Q).

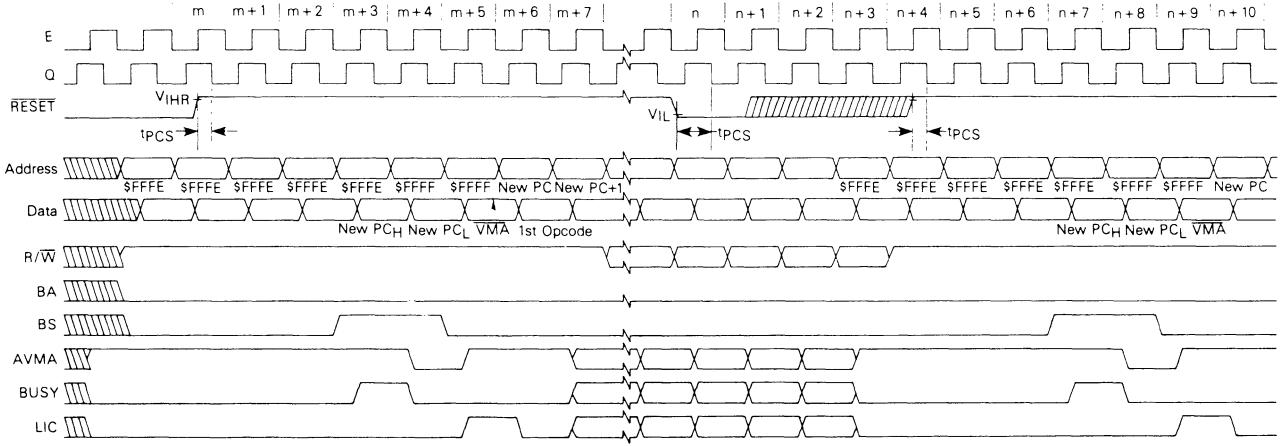
MPU State		MPU State Definition
BA	BS	
0	0	Normal (Running)
0	1	Interrupt or Reset Acknowledge
1	0	Sync Acknowledge
1	1	Halt Acknowledge

Interrupt Acknowledge is indicated during both cycles of a hardware vector fetch ($\overline{RESET}$, $\overline{NMI}$, $\overline{FIRQ}$, $\overline{IRQ}$, SWI, SWI2, SWI3). This signal, plus decoding of the lower four address lines, can provide the user with an indication of which interrupt level is being serviced and allow vectoring by device. See Table 1.

TABLE 1 – MEMORY MAP FOR INTERRUPT VECTORS

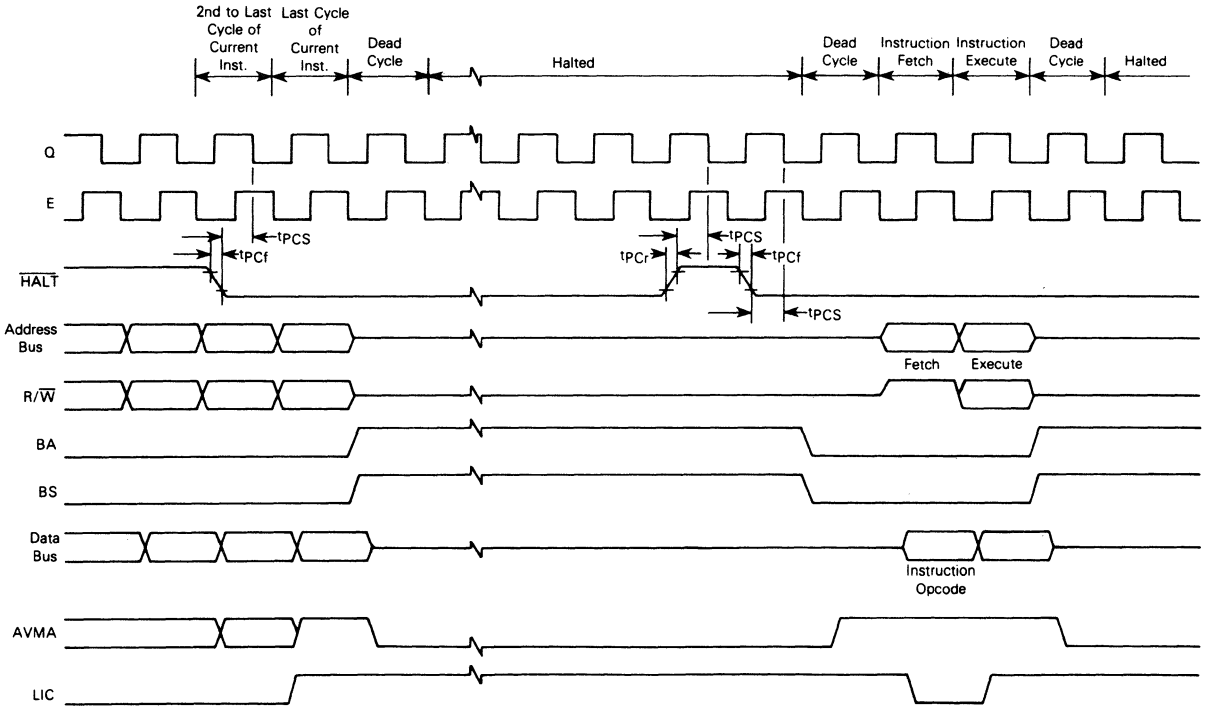
Memory Map For Vector Locations		Interrupt Vector Description
MS	LS	
FFFE	FFFF	$\overline{RESET}$
FFFC	FFFD	$\overline{NMI}$
FFFA	FFFB	SWI
FFF8	FFF9	$\overline{IRQ}$
FFF6	FFF7	$\overline{FIRQ}$
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reserved

FIGURE 6 — $\overline{\text{RESET}}$ TIMING



NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 7 — $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION TIMING FOR SYSTEM DEBUG



NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

Sync Acknowledge is indicated while the MPU is waiting for external synchronization on an interrupt line.

Halt Acknowledge is indicated when the EF6809E is in a halt condition.

NON MASKABLE INTERRUPT ($\overline{NMI}$)*

A negative transition on this input requests that a non-maskable interrupt sequence be generated. A non-maskable interrupt cannot be inhibited by the program and also has a higher priority than $\overline{FIRQ}$, $\overline{IRQ}$, or software interrupts. During recognition of an $\overline{NMI}$, the entire machine state is saved on the hardware stack. After reset, an $\overline{NMI}$ will not be recognized until the first program load of the hardware stack pointer (S). The pulse width of $\overline{NMI}$ low must be at least one E cycle. If the $\overline{NMI}$ input does not meet the minimum set up with respect to Q, the interrupt will not be recognized until the next cycle. See Figure 8.

FAST-INTERRUPT REQUEST ($\overline{FIRQ}$)*

A low level on this input pin will initiate a fast interrupt sequence, provided its mask bit (F) in the CC is clear. This sequence has priority over the standard interrupt request ($\overline{IRQ}$) and is fast in the sense that it stacks only the contents of the condition code register and the program counter. The interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 9.

INTERRUPT REQUEST ($\overline{IRQ}$)*

A low level input on this pin will initiate an interrupt request sequence provided the mask bit (I) in the CC is clear. Since $\overline{IRQ}$ stacks the entire machine state, it provides a slower response to interrupts than $\overline{FIRQ}$. $\overline{IRQ}$ also has a lower priority than $\overline{FIRQ}$. Again, the interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 8.

CLOCK INPUTS E, Q

E and Q are the clock signals required by the EF6809E. Q must lead E; that is, a transition on Q must be followed by a similar transition on E after a minimum delay. Addresses will be valid from the MPU, t_{AD} after the falling edge of E, and data will be latched from the bus by the falling edge of E. While the Q input is fully TTL compatible, the E input directly drives internal MOS circuitry and, thus, requires a high level above normal TTL levels. This approach minimizes clock skew inherent with an internal buffer. Refer to **BUS TIMING CHARACTERISTICS** for E and Q and to Figure 10 which shows a simple clock generator for the EF6809E.

BUSY

BUSY will be high for the read and modify cycles of a read-modify-write instruction and during the access of the first byte of a double-byte operation (e.g., LDX, STD, ADDD). BUSY is also high during the first byte of any indirect or other vector fetch (e.g., jump extended, SWI indirect, etc.).

In a multiprocessor system, BUSY indicates the need to defer the rearbitration of the next bus cycle to insure the integrity of the above operations. This difference provides the indivisible memory access required for a "test-and-set" primitive, using any one of several read-modify-write instructions.

BUSY does not become active during PSH or PUL operations. A typical read-modify-write instruction (ASL) is shown in Figure 11. Timing information is given in Figure 12. BUSY is valid t_{CD} after the rising edge of Q.

AVMA

AVMA is the advanced VMA signal and indicates that the MPU will use the bus in the following bus cycle. The predictive nature of the AVMA signal allows efficient shared-bus multiprocessor systems. AVMA is low when the MPU is in either a $\overline{HALT}$ or SYNC state. AVMA is valid t_{CD} after the rising edge of Q.

LIC

LIC (last instruction cycle) is high during the last cycle of every instruction, and its transition from high to low will indicate that the first byte of an opcode will be latched at the end of the present bus cycle. LIC will be high when the MPU is halted at the end of an instruction (i.e., not in CWAI or $\overline{RESET}$), in sync state, or while stacking during interrupts. LIC is valid t_{CD} after the rising edge of Q.

TSC

TSC (three-state control) will cause MOS address, data, and R/$\overline{W}$ buffers to assume a high-impedance state. The control signals (BA, BS, BUSY, AVMA, and LIC) will not go to the high-impedance state. TSC is intended to allow a single bus to be shared with other bus masters (processors or DMA controllers).

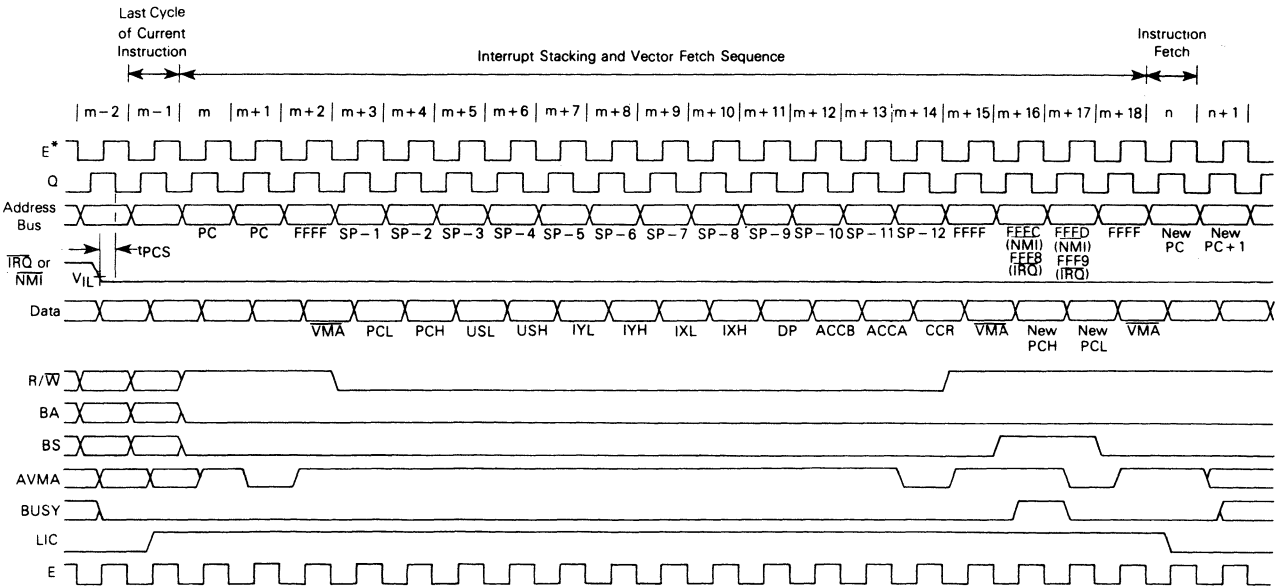
While E is low, TSC controls the address buffers and R/$\overline{W}$ directly. The data bus buffers during a write operation are in a high-impedance state until Q rises at which time, if TSC is true, they will remain in a high-impedance state. If TSC is held beyond the rising edge of E, then it will be internally latched, keeping the bus drivers in a high-impedance state for the remainder of the bus cycle. See Figure 13.

MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function. This sequence begins after $\overline{RESET}$ and is repeated indefinitely unless altered by a special instruction or hardware occurrence. Software instructions that alter normal MPU operation are: SWI, SWI2, SWI3, CWAI, RTI, and SYNC. An interrupt or $\overline{HALT}$ input can also alter the normal execution of instructions. Figure 14 is the flowchart for the EF6809E.

* $\overline{NMI}$, $\overline{FIRQ}$, and $\overline{IRQ}$ requests are sampled on the falling edge of Q. One cycle is required for synchronization before these interrupts are recognized. The pending interrupt(s) will not be serviced until completion of the current instruction unless a SYNC or CWAI condition is present. If $\overline{IRQ}$ and $\overline{FIRQ}$ do not remain low until completion of the current instruction, they may not be recognized. However, $\overline{NMI}$ is latched and need only remain low for one cycle. No interrupts are recognized or latched between the falling edge of $\overline{RESET}$ and the rising edge of BS indicating $\overline{RESET}$ acknowledge. See $\overline{RESET}$ sequence in the MPU flowchart in Figure 14.

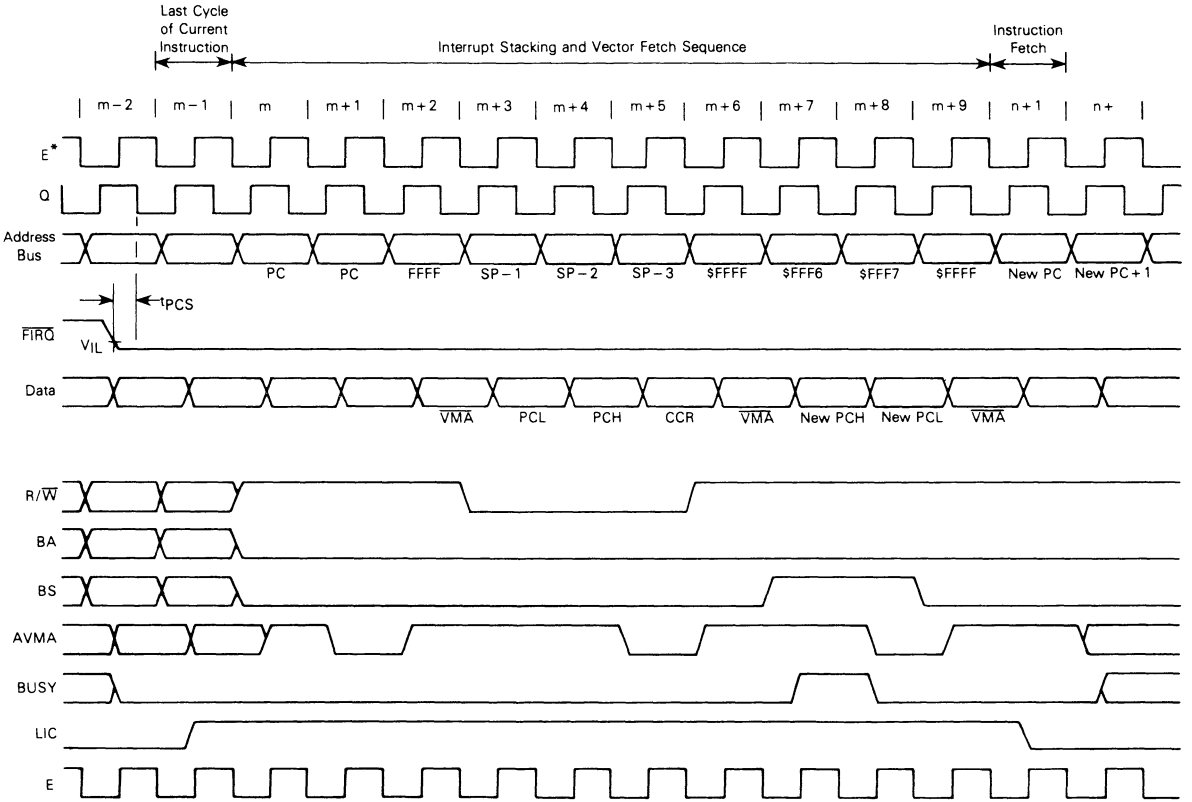
FIGURE 8 – $\overline{IRQ}$ AND $\overline{NMI}$ INTERRUPT TIMING



* E clock shown for reference only.

NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

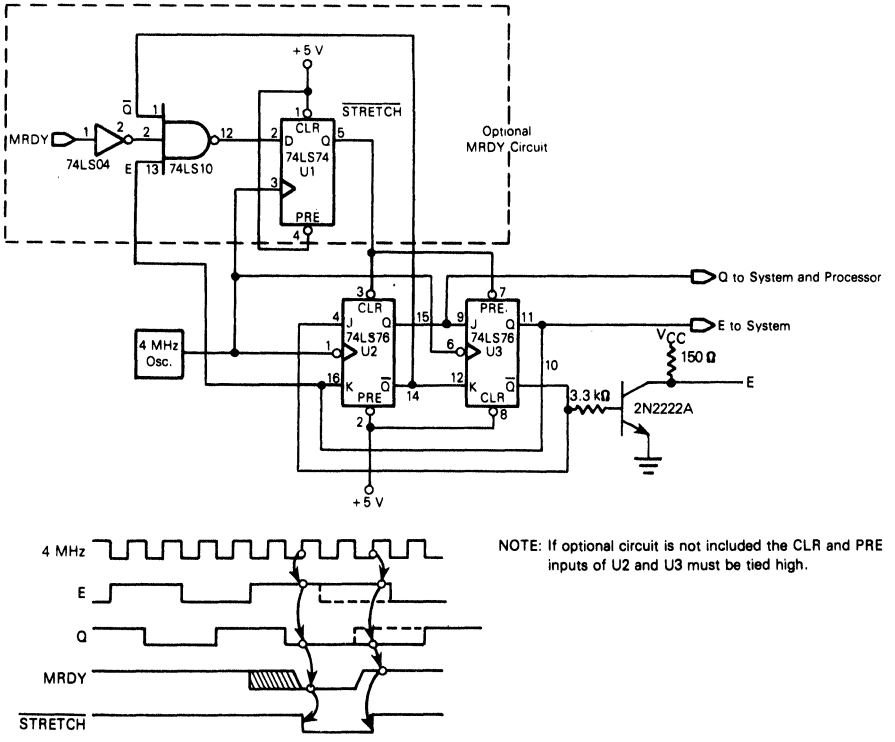
FIGURE 9 — $\overline{FIRQ}$ INTERRUPT TIMING



* E clock shown for reference only.

NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 10 – CLOCK GENERATOR



NOTE: If optional circuit is not included the CLR and PRE inputs of U2 and U3 must be tied high.

FIGURE 11 – READ-MODIFY-WRITE INSTRUCTION EXAMPLE (ASL EXTENDED INDIRECT)

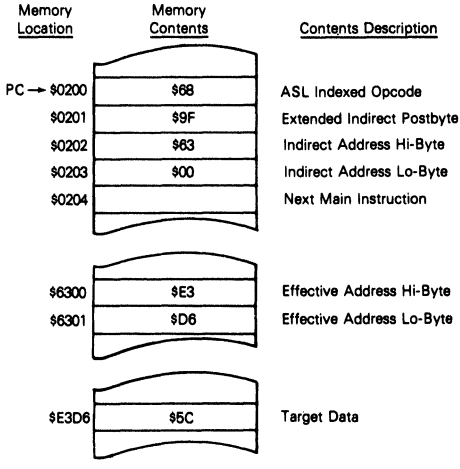


FIGURE 12 — BUSY TIMING

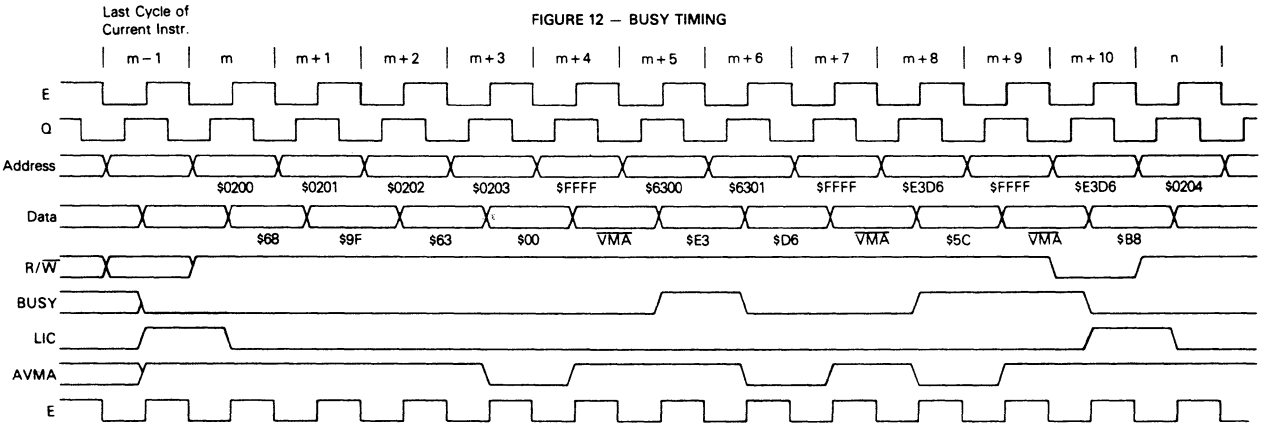
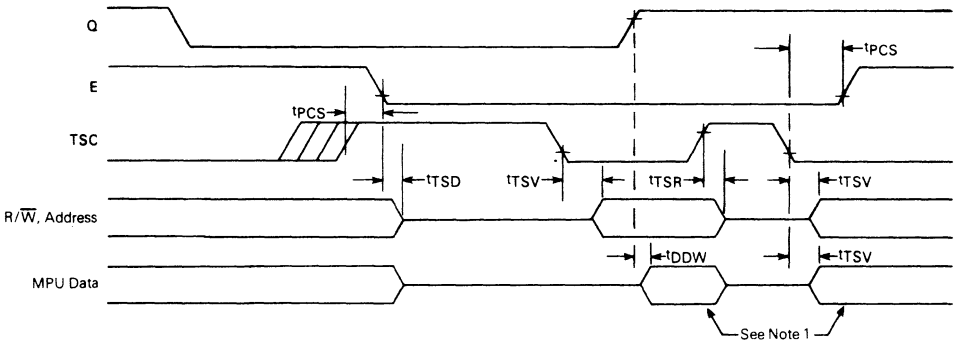


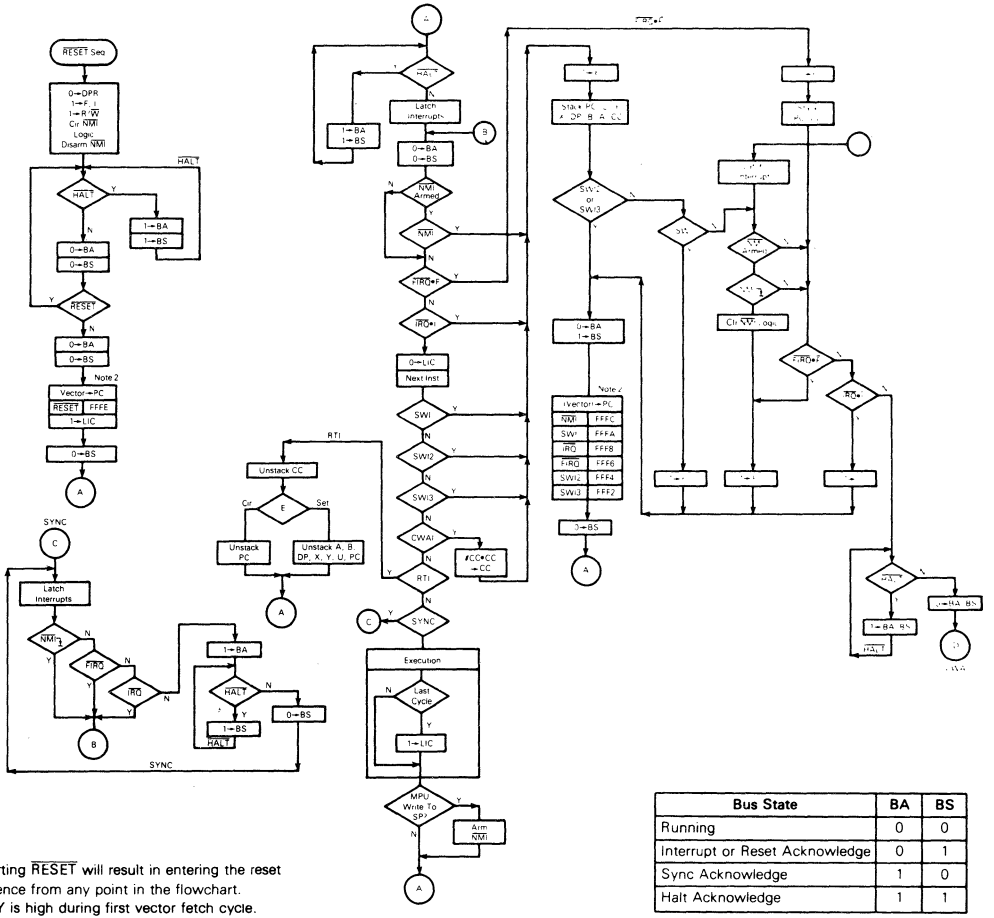
FIGURE 13 — TSC TIMING



NOTES:

1. Data will be asserted by the MPU only during the interval while R/$\overline{W}$ is low and (E or Q) is high. A composite bus cycle is shown to give most cases of timing.
2. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 14 – FLOWCHART FOR EF6809E INSTRUCTIONS



Bus State	BA	BS
Running	0	0
Interrupt or Reset Acknowledge	0	1
Sync Acknowledge	1	0
Halt Acknowledge	1	1

NOTES: 1. Asserting $\overline{\text{RESET}}$ will result in entering the reset sequence from any point in the flowchart.
2. BUSY is high during first vector fetch cycle.

ADDRESSING MODES

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The EF6809E has the most complete set of addressing modes available on any microcomputer today. For example, the EF6809E has 59 basic instructions; however, it recognizes 1464 different variations of instructions and addressing modes. The addressing modes support modern programming techniques. The following addressing modes are available on the EF6809E :

- Inherent (Includes Accumulator)
- Immediate
- Extended
  - Extended Indirect
- Direct
- Register
- Indexed
  - Zero-Offset
  - Constant Offset
  - Accumulator Offset
  - Auto Increment/Decrement
  - Indexed Indirect
- Relative
  - Short/Long Relative Branching
  - Program Counter Relative Addressing

INHERENT (INCLUDES ACCUMULATOR)

In this addressing mode, the opcode of the instruction contains all the address information necessary. Examples of inherent addressing are: ABX, DAA, SWI, ASRA, and CLRB.

IMMEDIATE ADDRESSING

In immediate addressing, the effective address of the data is the location immediately following the opcode (i.e., the data to be used in the instruction immediately following the opcode of the instruction). The EF6809E uses both 8- and 16-bit immediate values depending on the size of argument specified by the opcode. Examples of instructions with immediate addressing are:

```
LDA   #$20
LDX   #$F000
LDY   #CAT
```

NOTE

signifies immediate addressing; $ signifies hexadecimal value to the EF6809 assembler.

EXTENDED ADDRESSING

In extended addressing, the contents of the two bytes immediately following the opcode fully specify the 16-bit effective address used by the instruction. Note that the address generated by an extended instruction defines an absolute address and is not position independent. Examples of extended addressing include:

```
LDA   CAT
STX   MOUSE
LDD   $2000
```

EXTENDED INDIRECT

As a special case of indexed addressing (discussed below), one level of indirection may be added to extended addressing. In extended indirect, the two bytes following the postbyte of an indexed instruction contain the address of the data.

```
LDA   [CAT]
LDX   [$FFFE]
STU   [DOG]
```

DIRECT ADDRESSING

Direct addressing is similar to extended addressing except that only one byte of address follows the opcode. This byte specifies the lower eight bits of the address to be used. The upper eight bits of the address are supplied by the direct page register. Since only one byte of address is required in direct addressing, this mode requires less memory and executes faster than extended addressing. Of course, only 256 locations (one page) can be accessed without redefining the contents of the DP register. Since the DP register is set to $00 on reset, direct addressing on the EF6809E is upward compatible with direct addressing on the 6800. Indirection is not allowed in direct addressing. Some examples of direct addressing are:

```
LDA   where DP = $00
LDB   where DP = $10
LDD   <CAT
```

NOTE

< is an assembler directive which forces direct addressing.

REGISTER ADDRESSING

Some opcodes are followed by a byte that defines a register or set of registers to be used by the instruction. This is called a postbyte. Some examples of register addressing are:

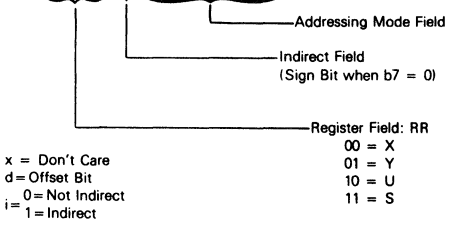
TFR	X, Y	Transfers X into Y
EXG	A, B	Exchanges A with B
PSHS	A, B, X, Y	Push Y, X, B and A onto S stack
PULU	X, Y, D	Pull D, X, and Y from U stack

INDEXED ADDRESSING

In all indexed addressing, one of the pointer registers (X, Y, U, S, and sometimes PC) is used in a calculation of the effective address of the operand to be used by the instruction. Five basic types of indexing are available and are discussed below. The postbyte of an indexed instruction specifies the basic type and variation of the addressing mode, as well as the pointer register to be used. Figure 15 lists the legal formats for the postbyte. Table 2 gives the assembler form and the number of cycles and bytes added to the basic values for indexed addressing for each variation.

FIGURE 15 – INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

Post-Byte Register Bit								Indexed Addressing Mode
7	6	5	4	3	2	1	0	
0	R	R	d	d	d	d	d	EA = ,R + 5 Bit Offset
1	R	R	0	0	0	0	0	,R+
1	R	R	i	0	0	0	1	,R+ +
1	R	R	0	0	0	1	0	,– R
1	R	R	i	0	0	1	1	,– – R
1	R	R	i	0	1	0	0	EA = ,R + 0 Offset
1	R	R	i	0	1	0	1	EA = ,R + ACCB Offset
1	R	R	i	0	1	1	0	EA = ,R + ACCA Offset
1	R	R	i	1	0	0	0	EA = ,R + 8 Bit Offset
1	R	R	i	1	0	0	1	EA = ,R + 16 Bit Offset
1	R	R	i	1	0	1	1	EA = ,R + D Offset
1	x	x	i	1	1	0	0	EA = ,PC + 8 Bit Offset
1	x	x	i	1	1	0	1	EA = ,PC + 16 Bit Offset
1	R	R	i	1	1	1	1	EA = [,Address]



ZERO-OFFSET INDEXED – In this mode, the selected pointer register contains the effective address of the data to be used by the instruction. This is the fastest indexing mode.

Examples are:

```
LDD   0, X
LDA   ,S
```

CONSTANT OFFSET INDEXED – In this mode, twos complement offset and the contents of one of the pointer registers are added to form the effective address of the operand. The pointer register's initial content is unchanged by the addition.

Three sizes of offset are available:

5-bit (– 16 to + 15)

8-bit (– 128 to + 127)

16-bit (– 32768 to + 32767)

The twos complement 5-bit offset is included in the post-byte and, therefore, is most efficient in use of bytes and cycles. The twos complement 8-bit offset is contained in a single byte following the postbyte. The twos complement 16-bit offset is in the two bytes following the postbyte. In most cases the programmer need not be concerned with the size of this offset since the assembler will select the optimal size automatically.

Examples of constant-offset indexing are:

```
LDA   23,X
LDX   – 2,S
LDY   300,X
LDU   CAT,Y
```

TABLE 2 – INDEXED ADDRESSING MODE

Type	Forms	Non Indirect				Indirect			
		Assembler Form	Postbyte Opcode	+ ~	+ #	Assembler Form	Postbyte Opcode	+ ~	+ #
Constant Offset From R (2s Complement Offsets)	No Offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
	5-Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8-Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16-Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R (2s Complement Offsets)	A Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
	B Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	,R+	1RR00000	2	0	not allowed			
	Increment By 2	,R+ +	1RR00001	3	0	[,R+ +]	1RR10001	6	0
	Decrement By 1	,– R	1RR00010	2	0	not allowed			
	Decrement By 2	,– – R	1RR00011	3	0	[,– – R]	1RR10011	6	0
Constant Offset From PC (2s Complement Offsets)	8-Bit Offset	n, PCR	1xx01100	1	1	[n, PCR]	1xx11100	4	1
	16-Bit Offset	n, PCR	1xx01101	5	2	[n, PCR]	1xx11101	8	2
Extended Indirect	16-Bit Address	–	–	–	–	[n]	10011111	5	2

R = X, Y, U or S
x = Don't Care

RR:
00 = X
01 = Y
10 = U
11 = S

$\overset{+}{\sim}$ and $\overset{+}{\#}$ indicate the number of additional cycles and bytes respectively for the particular indexing variation.

ACCUMULATOR-OFFSET INDEXED – This mode is similar to constant offset indexed except that the twos complement value in one of the accumulators (A, B, or D) and the contents of one of the pointer registers are added to form the effective address of the operand. The contents of both the accumulator and the pointer register are unchanged by the addition. The postbyte specifies which accumulator to use as an offset and no additional bytes are required. The advantage of an accumulator offset is that the value of the offset can be calculated by a program at run-time.

Some examples are:

```
LDA   B, Y
LDX   D, Y
LEAX  B, X
```

AUTO INCREMENT/DECREMENT INDEXED – In the auto increment addressing mode, the pointer register contains the address of the operand. Then, after the pointer register is used, it is incremented by one or two. This addressing mode is useful in stepping through tables, moving data, or creating software stacks. In auto decrement, the pointer register is decremented prior to use as the address of the data. The use of auto decrement is similar to that of auto increment, but the tables, etc., are scanned from the high to low addresses. The size of the increment/decrement can be either one or two to allow for tables of either 8- or 16-bit data to be accessed and is selectable by the programmer. The pre-decrement, post-increment nature of these modes allows them to be used to create additional software stacks that behave identically to the U and S stacks.

Some examples of the auto increment/decrement addressing modes are:

```
LDA   ,X+
STD   ,Y++
LDB   ,-Y
LDX   ,--S
```

Care should be taken in performing operations on 16-bit pointer registers (X, Y, U, S) where the same register is used to calculate the effective address.

Consider the following instruction:

STX 0,X++ (X initialized to 0)

The desired result is to store a zero in locations $0000 and $0001, then increment X to point to $0002. In reality, the following occurs:

0→temp	calculate the EA; temp is a holding register
X+2→X	perform auto increment
X→(temp)	do store operation

INDEXED INDIRECT

All of the indexing modes, with the exception of auto increment/decrement by one or a ±5-bit offset, may have an additional level of indirection specified. In indirect addressing, the effective address is contained at the location specified by the contents of the index register plus any offset. In the example below, the A accumulator is loaded indirectly using an effective address calculated from the index register and an offset.

```
Before Execution
  A = XX (don't care)
  X = $F000

$0100   LDA [$10,X]     EA is now $F010

$F010   $F1             $F150 is now the
$F011   $50             new EA

$F150   $AA

After Execution
  A = $AA (actual data loaded)
  X = $F000
```

All modes of indexed indirect are included except those which are meaningless (e.g., auto increment/decrement by 1 indirect). Some examples of indexed indirect are:

```
LDA   [,X]
LDD   [10,S]
LDA   [B,Y]
LDD   [,X++]
```

RELATIVE ADDRESSING

The byte(s) following the branch opcode is (are) treated as a signed offset which may be added to the program counter. If the branch condition is true, then the calculated address (PC + signed offset) is loaded into the program counter. Program execution continues at the new location as indicated by the PC; short (one byte offset) and long (two bytes offset) relative addressing modes are available. All of memory can be reached in long relative addressing as an effective address interpreted modulo 2^{16}. Some examples of relative addressing are:

```
        BEQ     CAT     (short)
        BGT     DOG     (short)
CAT     LBEQ    RAT     (long)
DOG     LBGT    RABBIT  (long)
        •
        •
        •
RAT     NOP
RABBIT  NOP
```

PROGRAM COUNTER RELATIVE

The PC can be used as the pointer register with 8- or 16-bit signed offsets. As in relative addressing, the offset is added to the current PC to create the effective address. The effective address is then used as the address of the operand or data. Program counter relative addressing is used for writing position independent programs. Tables related to a particular routine will maintain the same relationship after the routine is moved, if referenced relative to the program counter. Examples are:

```
LDA   CAT, PCR
LEAX  TABLE, PCR
```

Since program counter relative is a type of indexing, an additional level of indirection is available.

```
LDA   [CAT, PCR]
LDU   [DOG, PCR]
```

The instruction set of the EF6809E is similar to that of the EF6800 and is upward compatible at the source code level. The number of opcodes has been reduced from 72 to 59, but because of the expanded architecture and additional addressing modes, the number of available opcodes (with different addressing modes) has risen from 197 to 1464.

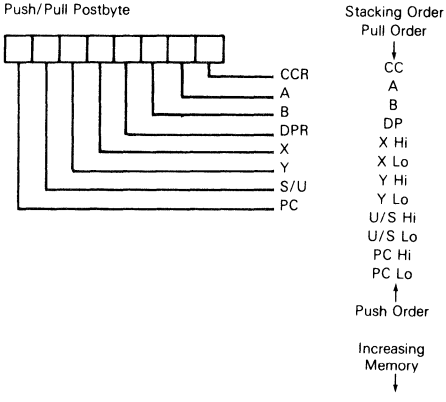
Some of the new instructions are described in detail below.

PSHU/PSHS

The push instructions have the capability of pushing onto either the hardware stack (S) or user stack (U) any single register or set of registers with a single instruction.

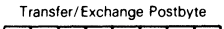
PULU/PULS

The pull instructions have the same capability of the push instruction, in reverse order. The byte immediately following the push or pull opcode determines which register or registers are to be pushed or pulled. The actual push/pull sequence is fixed; each bit defines a unique register to push or pull, as shown below.



TFR/EXG

Within the EF6809E, any register may be transferred to or exchanged with another of like size; i.e., 8-bit to 8-bit or 16-bit to 16-bit. Bits 4-7 of postbyte define the source register, while bits 0-3 represent the destination register. These are denoted as follows:



Register Field

0000 = D (A:B)	1000 = A
0001 = X	1001 = B
0010 = Y	1010 = CCR
0011 = U	1011 = DPR
0100 = S	
0101 = PC	

NOTE

All other combinations are undefined and INVALID.

LEAX/LEAY/LEAU/LEAS

The LEA (load effective address) works by calculating the effective address used in an indexed instruction and stores that address value, rather than the data at that address, in a pointer register. This makes all the features of the internal addressing hardware available to the programmer. Some of the implications of this instruction are illustrated in Table 3.

The LEA instruction also allows the user to access data and tables in a position independent manner. For example:

```
        LEAX    MSG1, PCR
        LBSR    PDATA (Print message routine)
        •
        •
MSG1    FCC     'MESSAGE'
```

This sample program prints: 'MESSAGE'. By writing MSG1, PCR, the assembler computes the distance between the present address and MSG1. This result is placed as a constant into the LEAX instruction which will be indexed from the PC value at the time of execution. No matter where the code is located when it is executed, the computed offset from the PC will put the absolute address of MSG1 into the X pointer register. This code is totally position independent.

The LEA instructions are very powerful and use an internal holding register (temp). Care must be exercised when using the LEA instructions with the auto increment and auto decrement addressing modes due to the sequence of internal operations. The LEA internal sequence is outlined as follows:

LEAa ,b+ (any of the 16-bit pointer registers X, Y, U, or S may be substituted for a and b.)

1. b → temp (calculate the EA)
2. b+1 → b (modify b, postincrement)
3. temp → a (load a)

LEAa ,−b

1. b−1 → temp (calculate EA with predecrement)
2. b−1 → b (modify b, predecrement)
3. temp → a (load a)

TABLE 3 — LEA EXAMPLES

Instruction	Operation	Comment
LEAX 10, X	X + 10 → X	Adds 5-Bit Constant 10 to X
LEAX 500, X	X + 500 → X	Adds 16-Bit Constant 500 to X
LEAY A, Y	Y + A → Y	Adds 8-Bit A Accumulator to Y
LEAY D, Y	Y + D → Y	Adds 16-Bit D Accumulator to Y
LEAU −10, U	U − 10 → U	Substracts 10 from U
LEAS −10, S	S − 10 → S	Used to Reserve Area on Stack
LEAS 10, S	S + 10 → S	Used to 'Clean Up' Stack
LEAX 5, S	S + 5 → X	Transfers As Well As Adds

Auto increment-by-two and auto decrement-by-two instructions work similarly. Note that LEAX ,X+ does not change X; however LEAX, – X does decrement X.LEAX 1,X should be used to increment X by one.

MUL

Multiplies the unsigned binary numbers in the A and B accumulator and places the unsigned result into the 16-bit D accumulator. This unsigned multiply also allows multiple-precision multiplications.

LONG AND SHORT RELATIVE BRANCHES

The EF6809E has the capability of program counter relative branching throughout the entire memory map. In this mode, if the branch is to be taken, the 8- or 16-bit signed offset is added to the value of the program counter to be used as the effective address. This allows the program to branch anywhere in the 64K memory map. Position independent code can be easily generated through the use of relative branching. Both short (8 bit) and long (16 bit) branches are available.

SYNC

After encountering a sync instruction, the MPU enters a sync state, stops processing instructions, and waits for an interrupt. If the pending interrupt is non-maskable ($\overline{NMI}$) or maskable ($\overline{FIRQ}$, $\overline{IRQ}$) with its mask bit (F or I) clear, the processor will clear the sync state and perform the normal interrupt stacking and service routine. Since $\overline{FIRQ}$ and $\overline{IRQ}$ are not edge-triggered, a low level with a minimum duration of three bus cycles is required to assure that the interrupt will be taken. If the pending interrupt is maskable ($\overline{FIRQ}$, $\overline{IRQ}$) with its mask bit (F or I) set, the processor will clear the sync state and continue processing by executing the next in-line instruction. Figure 16 depicts sync timing.

SOFTWARE INTERRUPTS

A software interrupt is an instruction which will cause an interrupt and its associated vector fetch. These software interrupts are useful in operating system calls, software debugging, trace operations, memory mapping, and software development systems. Three levels of SWI are available on this EF6809E and are prioritized in the following order: SWI, SWI2, SWI3.

16-BIT OPERATION

The EF6809E has the capability of processing 16-bit data. These instructions include loads, stores, compares, adds, subtracts, transfers, exchanges, pushes, and pulls.

CYCLE-BY-CYCLE OPERATION

The address bus cycle-by-cycle performance chart (Figure 16) illustrates the memory-access sequence corresponding to each possible instruction and addressing mode in the EF6809E. Each instruction begins with an opcode fetch. While that opcode is being internally decoded, the next program byte is always fetched. (Most instructions will use the next byte, so this technique considerably speeds throughput.) Next, the operation of each opcode will follow the flowchart. $\overline{VMA}$ is an indication of $FFFF_{16}$ on the address bus, R/$\overline{W}$ = 1 and BS = 0. The following examples illustrate the use of the chart.

Example 1: LBSR (Branch Taken)
Before Execution SP = F000

```
                  •
                  •
                  •
$8000             LBSR    CAT
                  •
                  •
                  •
$A000   CAT       •
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	R/$\overline{W}$	Description
1	8000	17	1	Opcode Fetch
2	8001	20	1	Offset High Byte
3	8002	00	1	Offset Low Byte
4	FFFF	*	1	$\overline{VMA}$ Cycle
5	FFFF	*	1	$\overline{VMA}$ Cycle
6	A000	*	1	Computed Branch Address
7	FFFF	*	1	$\overline{VMA}$ Cycle
8	EFFF	80	0	Stack High Order Byte of Return Address
9	EFFE	03	0	Stack Low Order Byte of Return Address

Example 2: DEC (Extended)

```
$8000   DEC     $A000
$A000   FCB     $80
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	R/$\overline{W}$	Description
1	8000	7A	1	Opcode Fetch
2	8001	A0	1	Operand Address, High Byte
3	8002	00	1	Operand Address, Low Byte
4	FFFF	*	1	$\overline{VMA}$ Cycle
5	A000	80	1	Read the Data
6	FFFF	*	1	$\overline{VMA}$ Cycle
7	FFFF	7F	0	Store the Decremented Data

* The data bus has the data at that particular address.

INSTRUCTION SET TABLES

The instructions of the EF6809E have been broken down into five different categories. They are as follows:

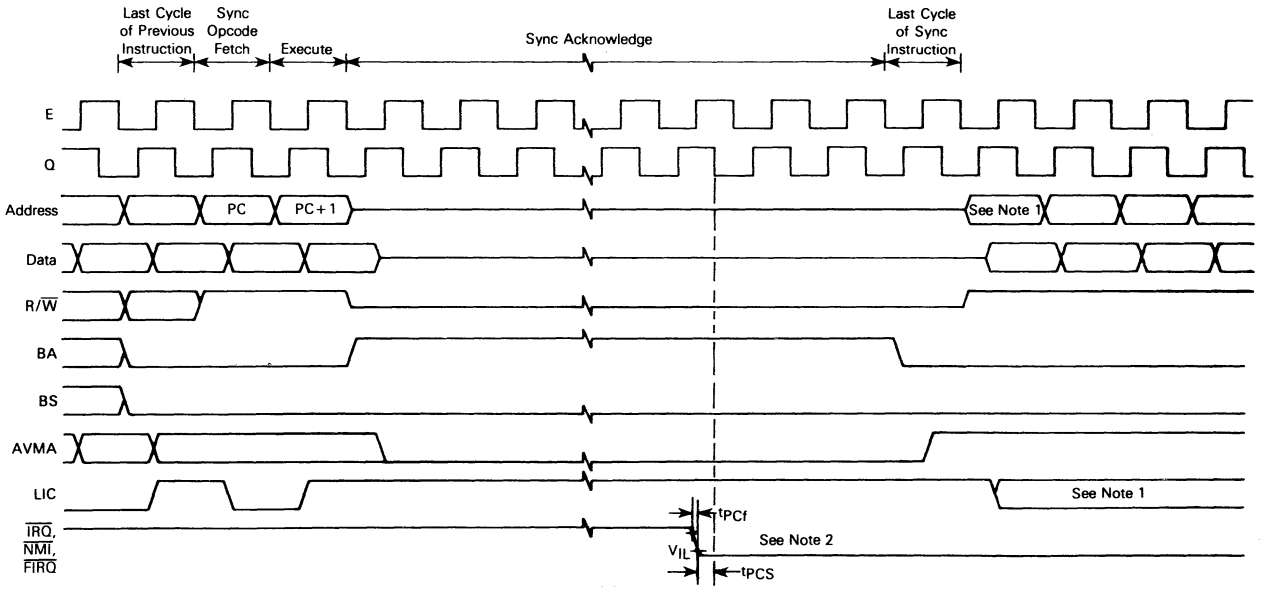
8-bit operation (Table 4)
16-bit operation (Table 5)
Index register/stack pointer instructions (Table 6)
Relative branches (long or short) (Table 7)
Miscellaneous instructions (Table 8)

Hexadecimal values for the instructions are given in Table 9.

PROGRAMMING AID

Figure 18 contains a compilation of data that will assist you in programming the EF6809E.

FIGURE 16 — SYNC TIMING

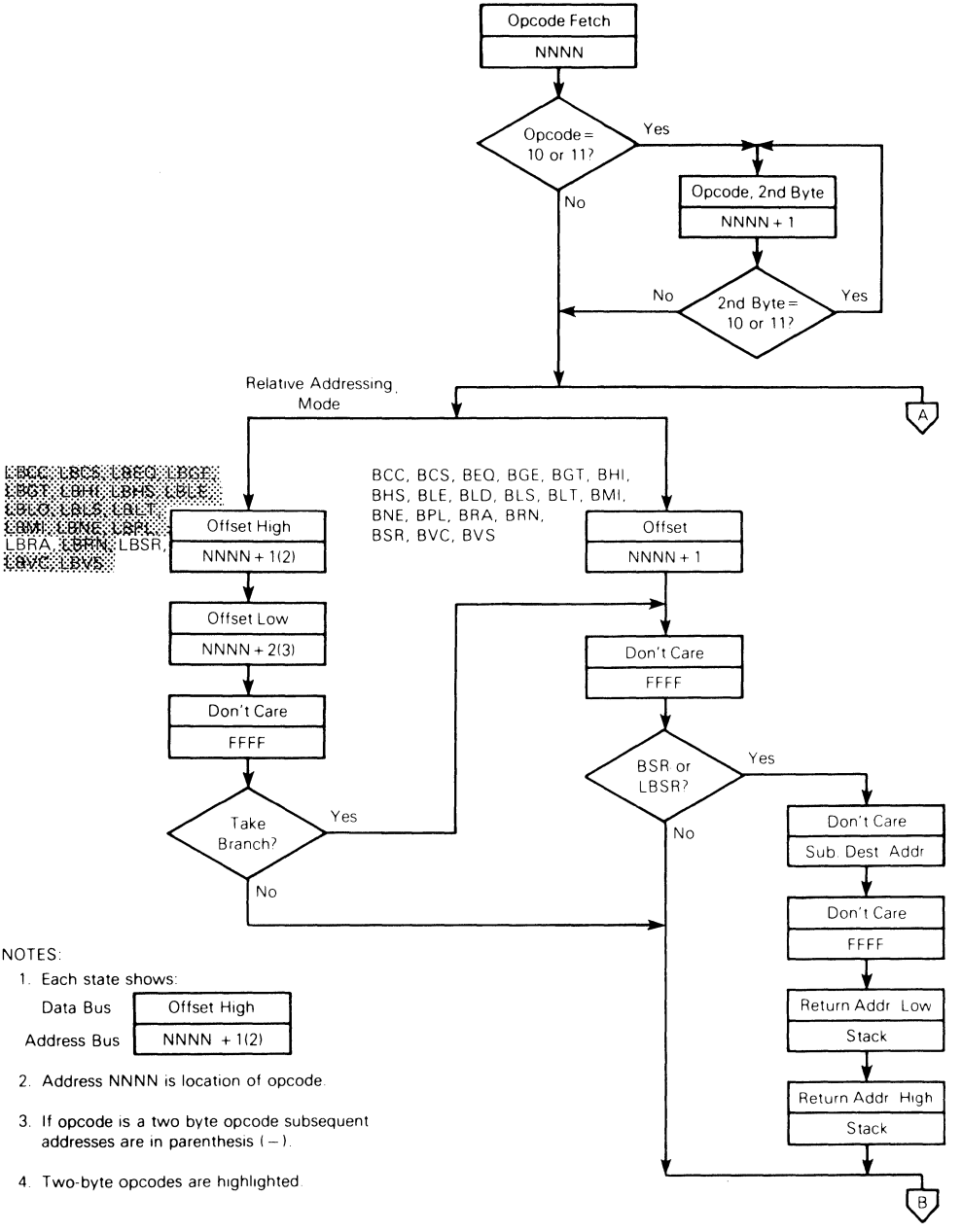


NOTES: 1. If the associated mask bit is set when the interrupt is requested, LIC will go low and this cycle will be an instruction fetch from address location PC+1. However, if the interrupt is accepted ($\overline{NMI}$ or an unmasked $\overline{FIRQ}$ or $\overline{IRQ}$) LIC will remain high and interrupt processing will start with this cycle as m on Figures 8 and 9 (Interrupt Timing).

2. If mask bits are clear, $\overline{IRQ}$ and $\overline{FIRQ}$ must be held low for three cycles to guarantee that interrupt will be taken, although only one cycle is necessary to bring the processor out of SYNC.

3. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 1 of 9)



NOTES:

1. Each state shows:

Data Bus	Offset High
Address Bus	NNNN + 1(2)

2. Address NNNN is location of opcode.
3. If opcode is a two byte opcode subsequent addresses are in parenthesis (–).
4. Two-byte opcodes are highlighted.

FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 2 of 9)

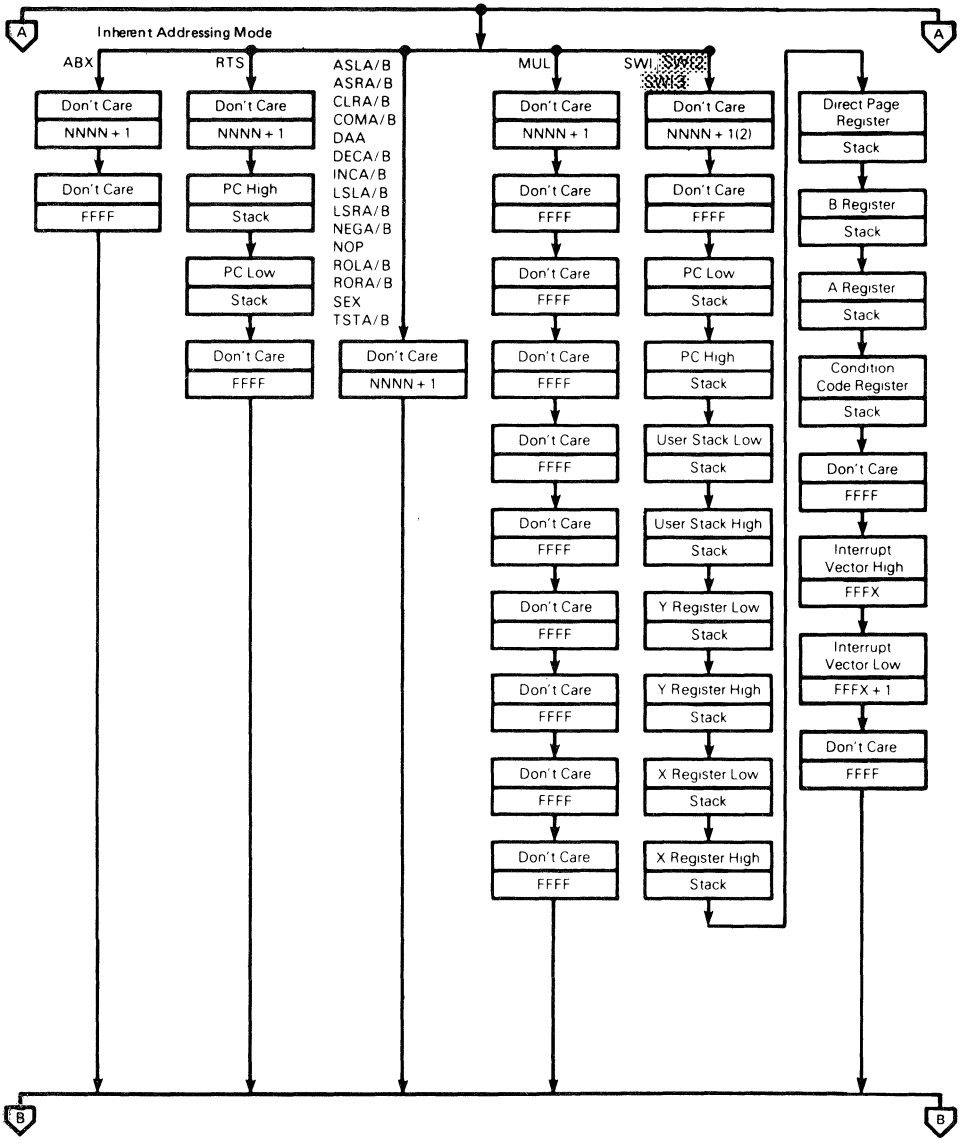


FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 3 of 9)

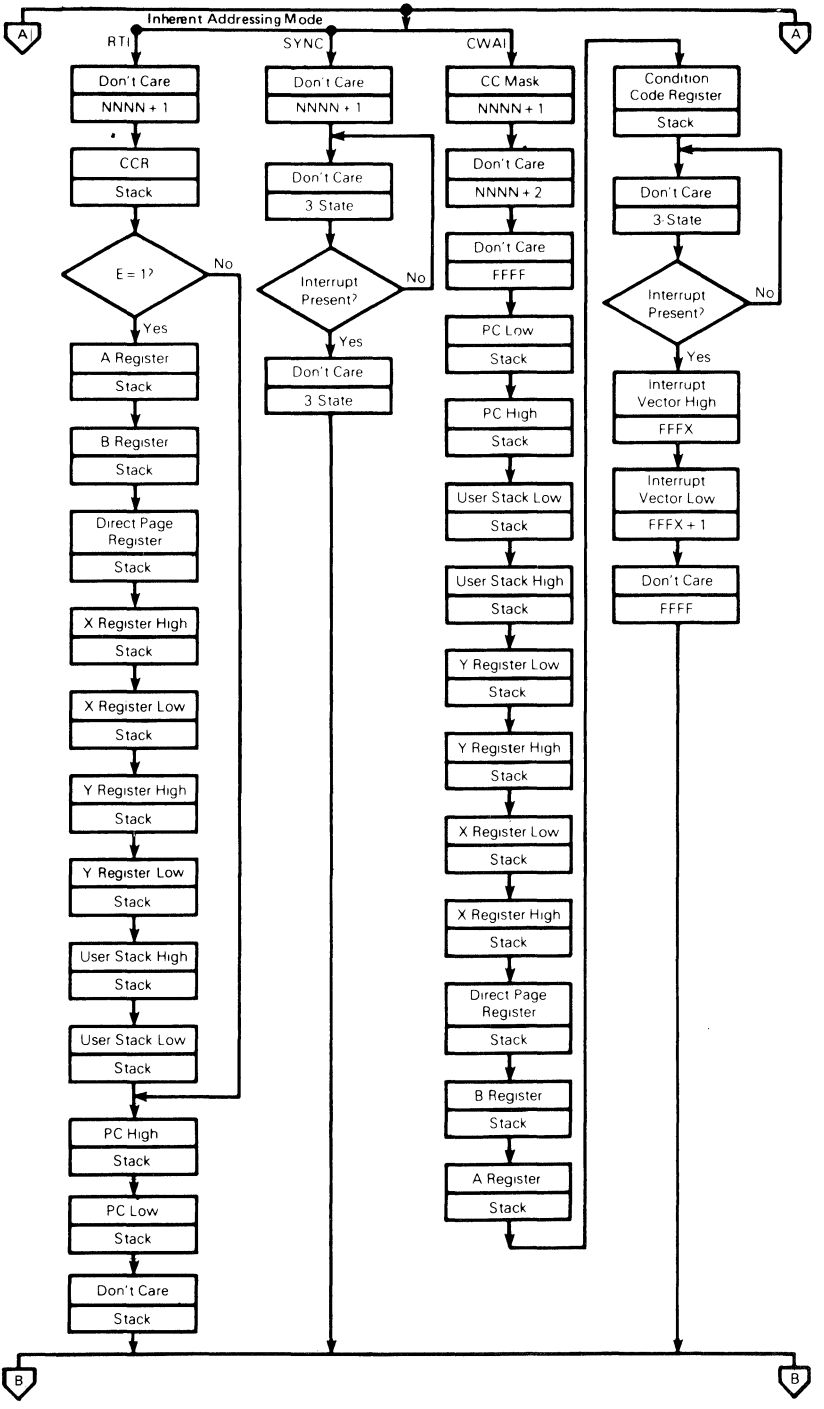


FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 4 of 9)

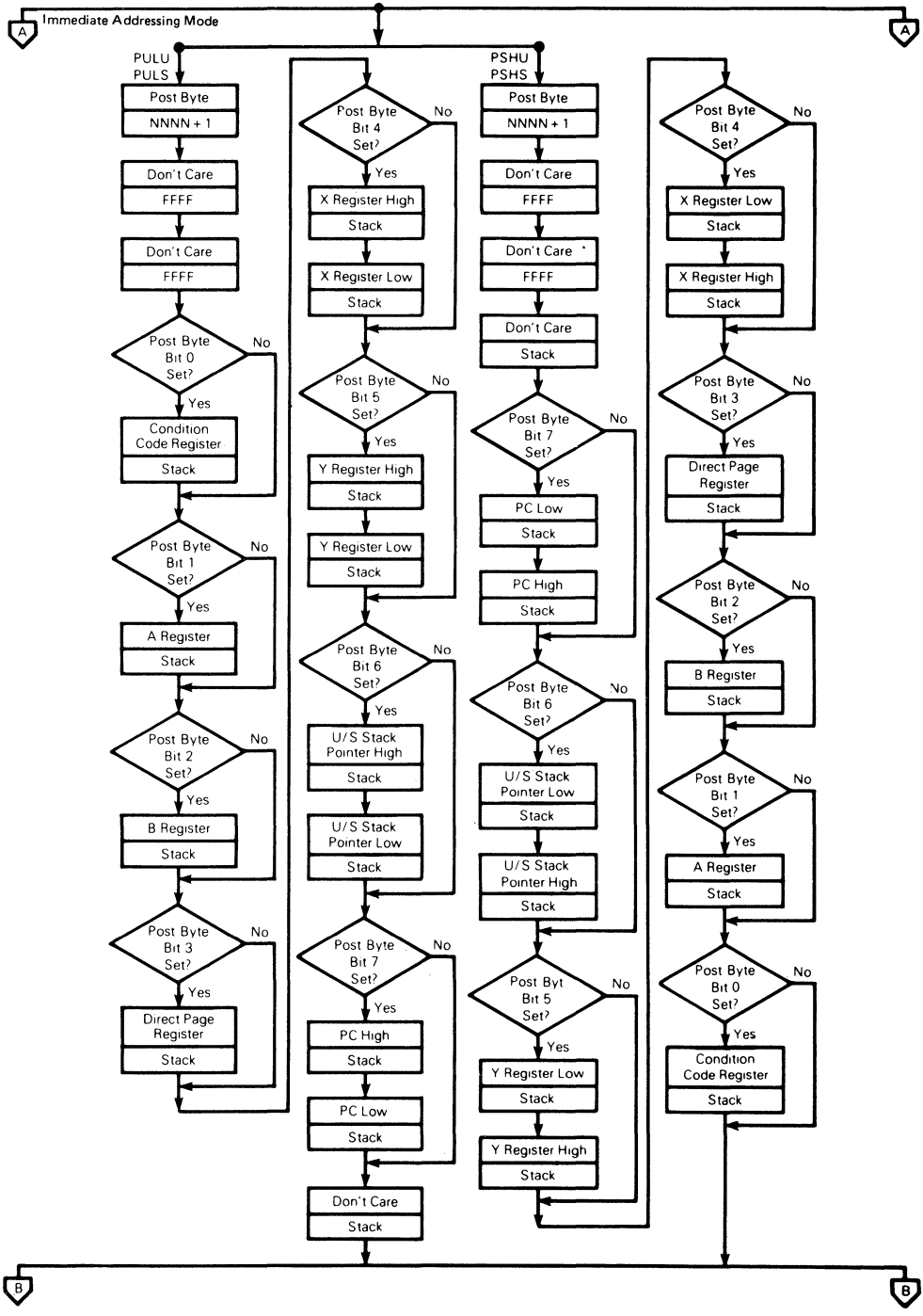


FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 5 of 9)

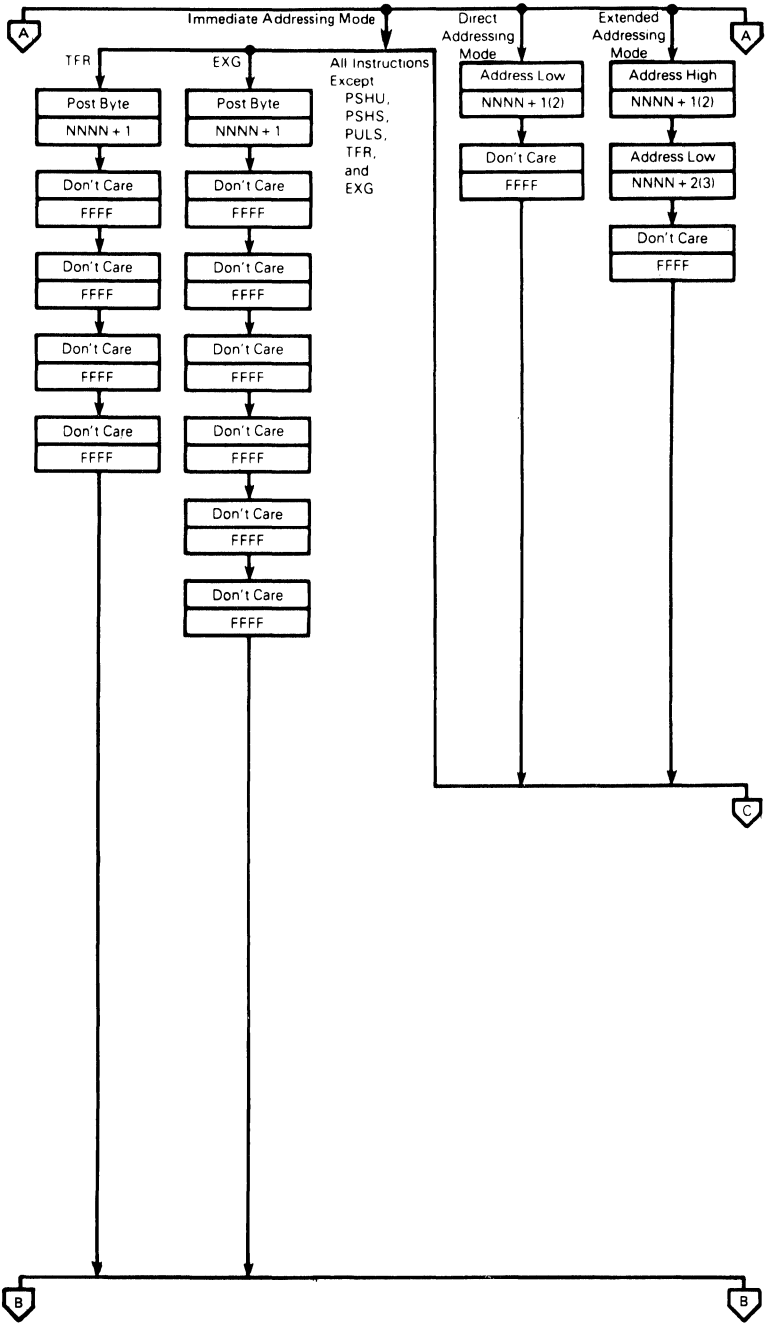
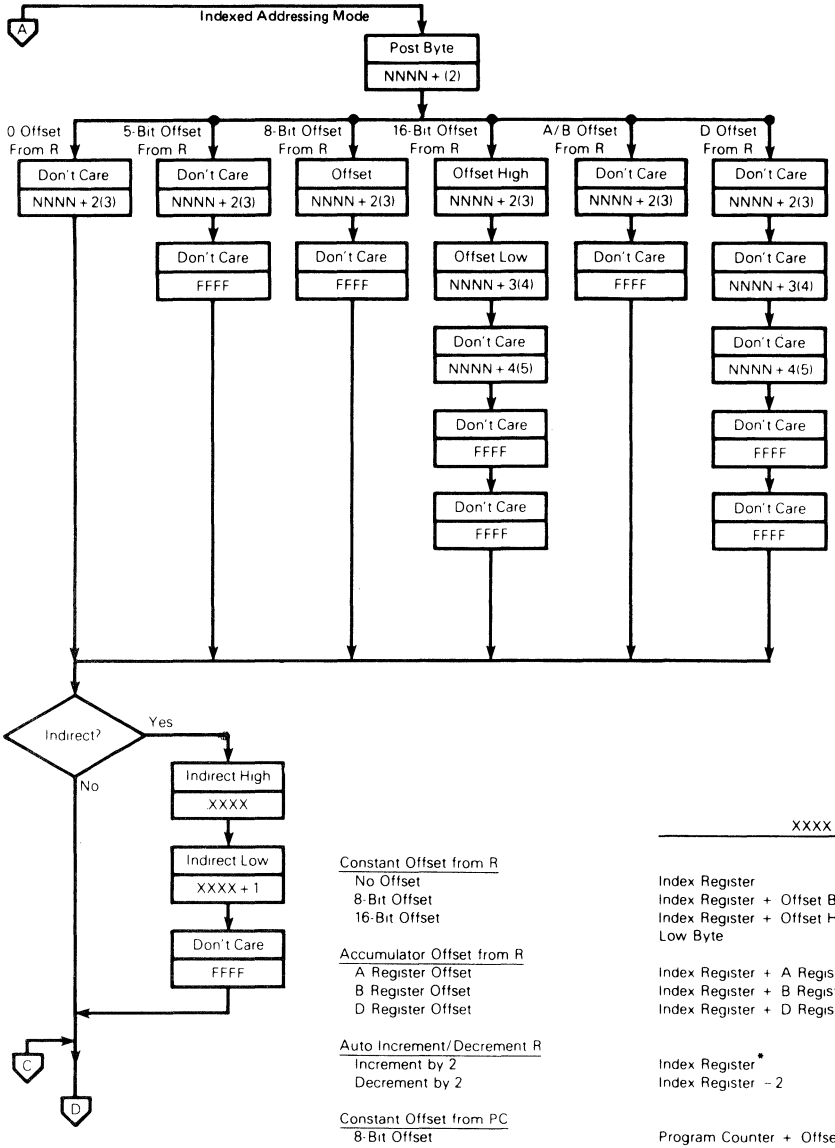


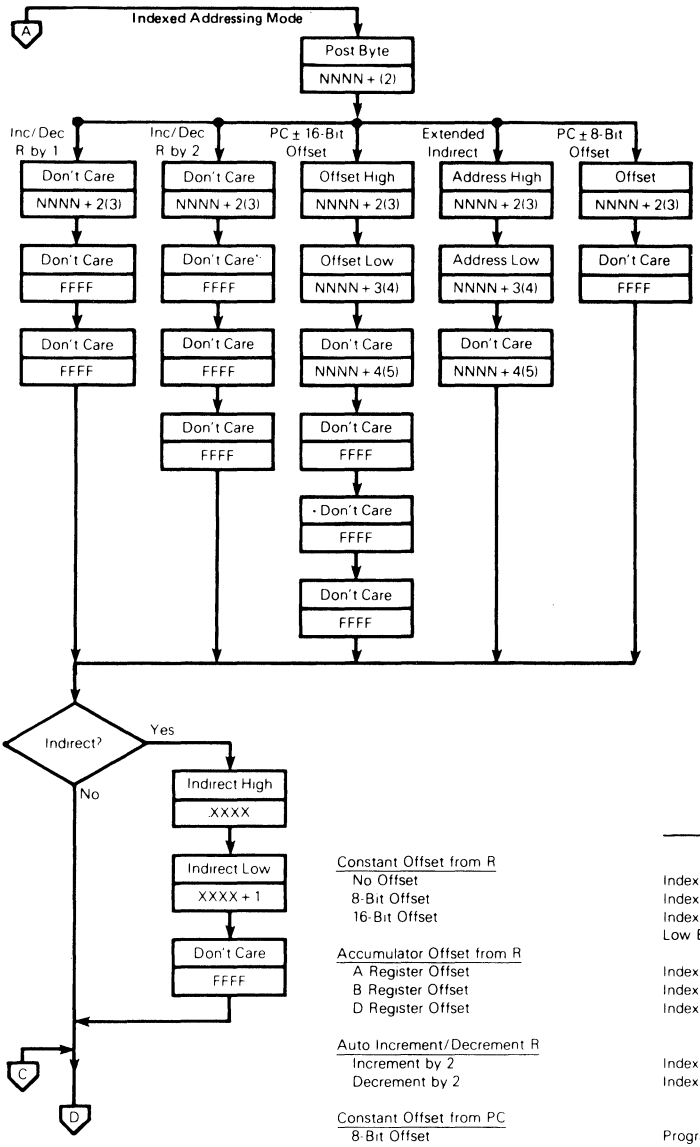
FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 6 of 9)



	XXXX
Constant Offset from R	
No Offset	Index Register
8-Bit Offset	Index Register + Offset Byte
16-Bit Offset	Index Register + Offset High Byte Offset Low Byte
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 2	Index Register*
Decrement by 2	Index Register – 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-bit Offset	Program Counter + Offset High Byte Offset Low Byte
Extended Indirect	
16-Bit Address	Address High Byte Addres Low Byte

* The index register is incremented following the indexed access

FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 7 of 9)



Constant Offset from R
- No Offset
- 8-Bit Offset
- 16-Bit Offset

Accumulator Offset from R
- A Register Offset
- B Register Offset
- D Register Offset

Auto Increment/Decrement R
- Increment by 2
- Decrement by 2

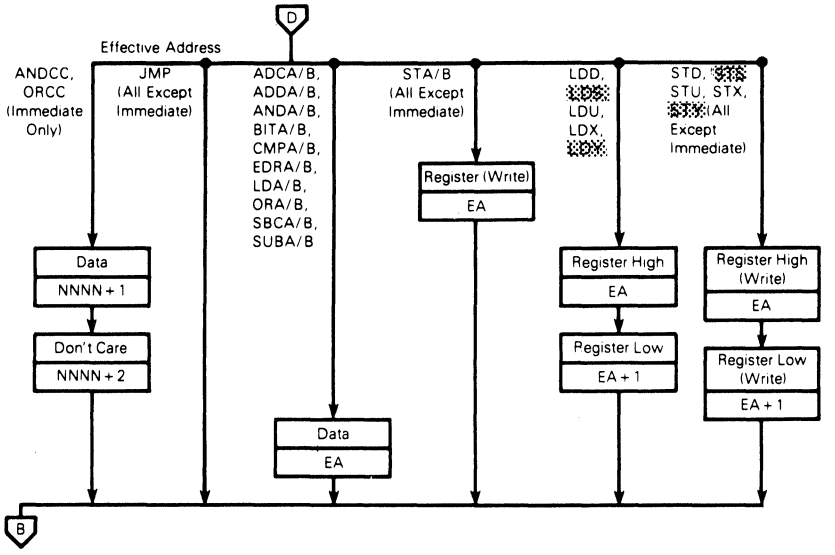
Constant Offset from PC
- 8-Bit Offset
- 16-bit Offset

Extended Indirect
- 16-Bit Address

* The index register is incremented following the indexed access

XXXX
Index Register
Index Register + Offset Byte
Index Register + Offset High Byte Offset Low Byte
Index Register + A Register
Index Register + B Register
Index Register + D Register
Index Register*
Index Register – 2
Program Counter + Offset Byte
Program Counter + Offset High Byte Offset Low Byte
Address High Byte Addres Low Byte

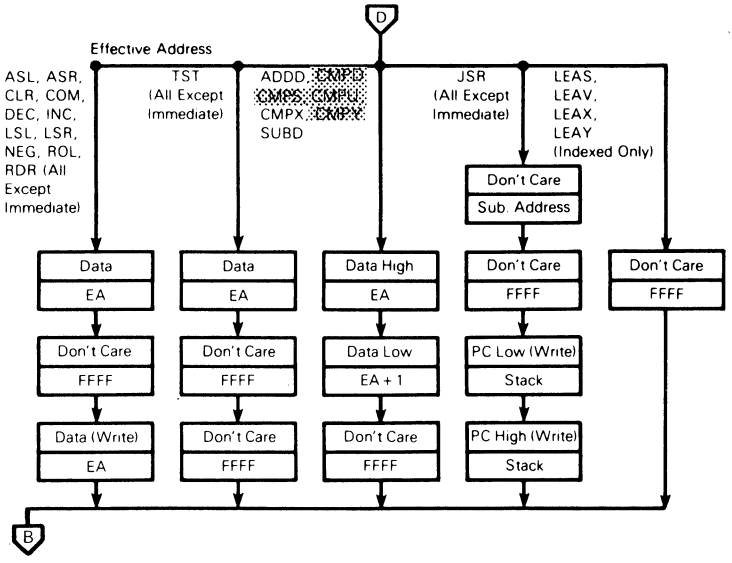
FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 8 of 9)



	Effective Address (EA)
Constant Offset from R	
No Offset	Index Register
5-Bit Offset	Index Register
8-Bit Offset	Index Register + Post Byte
16-Bit Offset	Index Register + Post Byte High, Post Byte Low
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 1	Index Register*
Increment by 2	Index Register*
Decrement by 1	Index Register – 1
Decrement by 2	Index Register – 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-Bit Offset	Program Counter + Offset High Byte, Offset Low Byte
Direct	Direct Page Register, Address Low
Extended	Address High, Address Low
Immediate	NNNN + 1

* The index register is incremented following the indexed access

FIGURE 17 – CYCLE-BY-CYCLE PERFORMANCE (Sheet 9 of 9)



	Effective Address (EA)
Constant Offset from R	
No Offset	Index Register
5-Bit Offset	Index Register
8-Bit Offset	Index Register + Post Byte
16-Bit Offset	Index Register + Post Byte High Post Byte Low
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 1	Index Register*
Increment by 2	Index Register*
Decrement by 1	Index Register – 1
Decrement by 2	Index Register – 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-Bit Offset	Program Counter + Offset High Byte Offset Low Byte
Direct	Direct Page Register Address Low
Extended	Address High Address Low
Immediate	NNNN + 1

* The index register is incremented following the indexed access

TABLE 4 — 8-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADCA, ADCB	Add memory to accumulator with carry
ADDA, ADDB	Add memory to accumulator
ANDA, ANDB	And memory with accumulator
ASL, ASLA, ASLB	Arithmetic shift of accumulator or memory left
ASR, ASRA, ASRB	Arithmetic shift of accumulator or memory right
BITA, BITB	Bit test memory with accumulator
CLR, CLRA, CLRB	Clear accumulator or memory location
CMPA, CMPB	Compare memory from accumulator
COM, COMA, COMB	Complement accumulator or memory location
DAA	Decimal adjust A accumulator
DEC, DECA, DECB	Decrement accumulator or memory location
EORA, EORB	Exclusive or memory with accumulator
EXG R1, R2	Exchange R1 with R2 (R1, R2 = A, B, CC, DP)
INC, INCA, INCB	Increment accumulator or memory location
LDA, LDB	Load accumulator from memory
LSL, LSLA, LSLB	Logical shift left accumulator or memory location
LSR, LSRA, LSRB	Logical shift right accumulator or memory location
MUL	Unsigned multiply (A × B → D)
NEG, NEGA, NEGB	Negate accumulator or memory
ORA, ORB	Or memory with accumulator
ROL, ROLA, ROLB	Rotate accumulator or memory left
ROR, RORA, RORB	Rotate accumulator or memory right
SBCA, SBCB	Subtract memory from accumulator with borrow
STA, STB	Store accumulator to memory
SUBA, SUBB	Subtract memory from accumulator
TST, TSTA, TSTB	Test accumulator or memory location
TFR R1, R2	Transfer R1 to R2 (R1, R2 = A, B, CC, DP)

NOTE: A, B, CC or DP may be pushed to (pulled from) either stack with PSHS, PSHU (PULS, PULU) instructions.

TABLE 5 — 16-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADDD	Add memory to D accumulator
CMPD	Compare memory from D accumulator
EXG D, R	Exchange D with X, Y, S, U or PC
LDD	Load D accumulator from memory
SEX	Sign Extend B accumulator into A accumulator
STD	Store D accumulator to memory
SUBD	Subtract memory from D accumulator
TFR D, R	Transfer D to X, Y, S, U or PC
TFR R, D	Transfer X, Y, S, U or PC to D

NOTE: D may be pushed (pulled) to either stack with PSHS, PSHU (PULS, PULU) instructions.

TABLE 6 — INDEX REGISTER/STACK POINTER INSTRUCTIONS

Instruction	Description
CMPS, CMPU	Compare memory from stack pointer
CMPX, CMPY	Compare memory from index register
EXG R1, R2	Exchange D, X, Y, S, U or PC with D, X, Y, S, U or PC
LEAS, LEAU	Load effective address into stack pointer
LEAX, LEAY	Load effective address into index register
LDS, LDU	Load stack pointer from memory
LDX, LDY	Load index register from memory
PSHS	Push A, B, CC, DP, D, X, Y, U, or PC onto hardware stack
PSHU	Push A, B, CC, DP, D, X, Y, S, or PC onto user stack
PULS	Pull A, B, CC, DP, D, X, Y, U or PC from hardware stack
PULU	Pull A, B, CC, DP, D, X, Y, S or PC from hardware stack
STS, STU	Store stack pointer to memory
STX, STY	Store index register to memory
TFR R1, R2	Transfer D, X, Y, S, U or PC to D, X, Y, S, U or PC
ABX	Add B accumulator to X (unsigned)

TABLE 7 – BRANCH INSTRUCTIONS

Instruction	Description
SIMPLE BRANCHES	
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BMI, LBMI	Branch if minus
BPL, LBPL	Branch if plus
BCS, LBCS	Branch if carry set
BCC, LBCC	Branch if carry clear
BVS, LBVS	Branch if overflow set
BVC, LBVC	Branch if overflow clear
SIGNED BRANCHES	
BGT, LBGT	Branch if greater (signed)
BVS, LBVS	Branch if invalid 2's complement result
BGE, LBGE	Branch if greater than or equal (signed)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLE, LBLE	Branch if less than or equal (signed)
BVC, LBVC	Branch if valid 2's complement result
BLT, LBLT	Branch if less than (signed)
UNSIGNED BRANCHES	
BHI, LBHI	Branch if higher (unsigned)
BCC, LBCC	Branch if higher or same (unsigned)
BHS, LBHS	Branch if higher or same (unsigned)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLS, LBLS	Branch if lower or same (unsigned)
BCS, LBCS	Branch if lower (unsigned)
BLO, LBLO	Branch if lower (unsigned)
OTHER BRANCHES	
BSR, LBSR	Branch to subroutine
BRA, LBRA	Branch always
BRN, LBRN	Branch never

TABLE 8 – MISCELLANEOUS INSTRUCTIONS

Instruction	Description
ANDCC	AND condition code register
CWAI	AND condition code register, then wait for interrupt
NOP	No operation
ORCC	OR condition code register
JMP	Jump
JSR	Jump to subroutine
RTI	Return from interrupt
RTS	Return from subroutine
SWI, SWI2, SWI3	Software interrupt (absolute indirect)
SYNC	Synchronize with interrupt line

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES

OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#
00	NEG	Direct	6	2	30	LEAX	Indexed	4+	2+	60	NEG	Indexed	6+	2+
01	*				31	LEAY		4+	2+	61	*			
02	*				32	LEAS		4+	2+	62	*			
03	COM		6	2	33	LEAU		4+	2+	63	COM		6+	2+
04	LSR		6	2	34	PSHS	Immed	5+	2	64	LSR		6+	2+
05	*				35	PULS	Immed	5+	2	65	*			
06	ROR		6	2	36	PSHU	Immed	5+	2	66	ROR		6+	2+
07	ASR		6	2	37	PULU	Immed	5+	2	67	ASR		6+	2+
08	ASL, LSL		6	2	38	*		–		68	ASL, LSL		6+	2+
09	ROL		6	2	39	RTS	Inherent	5	1	69	ROL		6+	2+
0A	DEC		6	2	3A	ABX		3	1	6A	DEC		6+	2+
0B	*				3B	RTI		6/15	1	6B	*			
0C	INC		6	2	3C	CWAI		≥20	2	6C	INC		6+	2+
0D	TST		6	2	3D	MUL	Inherent	11	1	6D	TST		6+	2+
0E	JMP		3	2	3E	*		–		6E	JMP		3+	2+
0F	CLR	Direct	6	2	3F	SWI	Inherent	19	1	6F	CLR	Indexed	6+	2+
10	Page 2	–	–	–	40	NEGA	Inherent	2	1	70	NEG	Extended	7	3
11	Page 3	–	–	–	41	*				71	*			
12	NOP	Inherent	2	1	42	*				72	*			
13	SYNC	Inherent	≥4	1	43	COMA		2	1	73	COM		7	3
14	*				44	LSRA		2	1	74	LSR		7	3
15	*				45	*				75	*			
16	LBRA	Relative	5	3	46	RORA		2	1	76	ROR		7	3
17	LBSR	Relative	9	3	47	ASRA		2	1	77	ASR		7	3
18	*				48	ASLA, LSLA		2	1	78	ASL, LSL		7	3
19	DAA	Inherent	2	1	49	ROLA		2	1	79	ROL		7	3
1A	ORCC	Immed	3	2	4A	DECA		2	1	7A	DEC		7	3
1B	*	–			4B	*				7B	*			
1C	ANDCC	Immed	3	2	4C	INCA		2	1	7C	INC		7	3
1D	SEX	Inherent	2	1	4D	TSTA		2	1	7D	TST		7	3
1E	EXG	Immed	8	2	4E	*				7E	JMP		4	3
1F	TFR	Immed	6	2	4F	CLRA	Inherent	2	1	7F	CLR	Extended	7	3
20	BRA	Relative	3	2	50	NEGB	Inherent	2	1	80	SUBA	Immed	2	2
21	BRN		3	2	51	*				81	CMPA		2	2
22	BHI		3	2	52	*				82	SBCA		2	2
23	BLS		3	2	53	COMB		2	1	83	SUBD		4	3
24	BHS, BCC		3	2	54	LSRB		2	1	84	ANDA		2	2
25	BLO, BCS		3	2	55	*				85	BITA		2	2
26	BNE		3	2	56	RORB		2	1	86	LDA		2	2
27	BEQ		3	2	57	ASRB		2	1	87	*			
28	BVC		3	2	58	ASLB, LSLB		2	1	88	EORA		2	2
29	BVS		3	2	59	ROLB		2	1	89	ADCA		2	2
2A	BPL		3	2	5A	DECB		2	1	8A	ORA		2	2
2B	BMI		3	2	5B	*				8B	ADDA		2	2
2C	BGE		3	2	5C	INCB		2	1	8C	CMPX	Immed	4	3
2D	BLT		3	2	5D	TSTB		2	1	8D	BSR	Relative	7	2
2E	BGT		3	2	5E	*				8E	LDX	Immed	3	3
2F	BLE	Relative	3	2	5F	CLRB	Inherent	2	1	8F	*			

LEGEND:

~ Number of MPU cycles (less possible push pull or indexed-mode cycles)

Number of program bytes

* Denotes unused opcode

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES (CONTINUED)

OP	Mnem	Mode	~	#
90	SUBA	Direct	4	2
91	CMPA		4	2
92	SBCA		4	2
93	SUBD		6	2
94	ANDA		4	2
95	BITA		4	2
96	LDA		4	2
97	STA		4	2
98	EORA		4	2
99	ADCA		4	2
9A	ORA		4	2
9B	ADDA		4	2
9C	CMPX		6	2
9D	JSR		7	2
9E	LDX		5	2
9F	STX	Direct	5	2
A0	SUBA	Indexed	4+	2+
A1	CMPA		4+	2+
A2	SBCA		4+	2+
A3	SUBD		6+	2+
A4	ANDA		4+	2+
A5	BITA		4+	2+
A6	LDA		4+	2+
A7	STA		4+	2+
A8	EORA		4+	2+
A9	ADCA		4+	2+
AA	ORA		4+	2+
AB	ADDA		4+	2+
AC	CMPX		6+	2+
AD	JSR		7+	2+
AE	LDX		5+	2+
AF	STX	Indexed	5+	2+
B0	SUBA	Extended	5	3
B1	CMPA		5	3
B2	SBCA		5	3
B3	SUBD		7	3
B4	ANDA		5	3
B5	BITA		5	3
B6	LDA		5	3
B7	STA		5	3
B8	EORA		5	3
B9	ADCA		5	3
BA	ORA		5	3
BB	ADDA		5	3
BC	CMPX		7	3
BD	JSR		8	3
BE	LDX		6	3
BF	STX	Extended	6	3

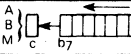
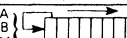
NOTE: All unused opcodes are both undefined and illegal

OP	Mnem	Mode	~	#
C0	SUBB	Immed	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB	Immed	2	2
C6	LDB	Immed	2	2
C7	*			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	*			
CE	LDU	Immed	3	3
CF	*			
D0	SUBB	Direct	4	2
D1	CMPB		4	2
D2	SBCB		4	2
D3	ADDD		6	2
D4	ANDB		4	2
D5	BITB		4	2
D6	LDB		4	2
D7	STB		4	2
D8	EORB		4	2
D9	ADCB		4	2
DA	ORB		4	2
DB	ADDB		4	2
DC	LDD		5	2
DD	STD		5	2
DE	LDU		5	2
DF	STU	Direct	5	2
E0	SUBB	Indexed	4+	2+
E1	CMPB		4+	2+
E2	SBCB		4+	2+
E3	ADDD		6+	2+
E4	ANDB		4+	2+
E5	BITB		4+	2+
E6	LDB		4+	2+
E7	STB		4+	2+
E8	EORB		4+	2+
E9	ADCB		4+	2+
EA	ORB		4+	2+
EB	ADDB		4+	2+
EC	LDD		5+	2+
ED	STD		5+	2+
EE	LDU		5+	2+
EF	STU	Indexed	5+	2+
F0	SUBB	Extended	5	3
F1	CMPB		5	3
F2	SBCB		5	3
F3	ADDD		7	3
F4	ANDB		5	3
F5	BITB		5	3
F6	LDB		5	3
F7	STB		5	3
F8	EORB		5	3
F9	ADCB		5	3
FA	ORB		5	3
FB	ADDB	Extended	5	3
FC	LDD	Extended	6	3
FD	STD		6	3
FE	LDU		6	3
FF	STU	Extended	6	3

Page 2 and 3 Machine Codes

OP	Mnem	Mode	~	#
1021	LBRN	Relative	5	4
1022	LBHI		5(6)	4
1023	LBLS		5(6)	4
1024	LBHS, LBCC		5(6)	4
1025	LBCS, LBLO		5(6)	4
1026	LBNE		5(6)	4
1027	LBEQ		5(6)	4
1028	LBVC		5(6)	4
1029	LBVS		5(6)	4
102A	LBPL		5(6)	4
102B	LBMI		5(6)	4
102C	LBGE		5(6)	4
102D	LBLT		5(6)	4
102E	LBGT		5(6)	4
102F	LBLE	Relative	5(6)	4
103F	SWI2	Inherent	20	2
1083	CMPD	Immed	5	4
108C	CMPY		5	4
108E	LDY	Immed	4	4
1093	CMPD	Direct	7	3
109C	CMPY		7	3
109E	LDY		6	3
109F	STY	Direct	6	3
10A3	CMPD	Indexed	7+	3+
10AC	CMPY		7+	3+
10AE	LDY		6+	3+
10AF	STY	Indexed	6+	3+
10B3	CMPD	Extended	8	4
10BC	CMPY		8	4
10BE	LDY		7	4
10BF	STY	Extended	7	4
10CE	LDS	Immed	4	4
10DE	LDS	Direct	6	3
10DF	STS	Direct	6	3
10EE	LDS	Indexed	6+	3+
10EF	STS	Indexed	6+	3+
10FE	LDS	Extended	7	4
10FF	STS	Extended	7	4
113F	SWI3	Inherent	20	2
1183	CMPU	Immed	5	4
118C	CMPS	Immed	5	4
1193	CMPU	Direct	7	3
119C	CMPS	Direct	7	3
11A3	CMPU	Indexed	7+	3+
11AC	CMPS	Indexed	7+	3+
11B3	CMPU	Extended	8	4
11BC	CMPS	Extended	8	4

FIGURE 18 — PROGRAMMING AID

Instruction	Forms	Addressing Modes: Immediate Op	Immediate ~	Immediate #	Direct Op	Direct ~	Direct #	Indexed Op	Indexed ~	Indexed #	Extended Op	Extended ~	Extended #	Inherent Op	Inherent ~	Inherent #	Description	5 H	3 N	2 Z	1 V	0 C
ABX														3A	3	1	B + X → X (Unsigned)	•	•	•	•	•
ADC	ADCA	89	2	2	99	4	2	A9	4 +	2 +	B9	5	3				A + M + C → A	↕	↕	↕	↕	↕
	ADCB	C9	2	2	D9	4	2	E9	4 +	2 +	F9	5	3				B + M + C → B	↕	↕	↕	↕	↕
ADD	ADDA	8B	2	2	9B	4	2	AB	4 +	2 +	BB	5	3				A + M → A	↕	↕	↕	↕	↕
	ADDB	CB	2	2	DB	4	2	EB	4 +	2 +	FB	5	3				B + M → B	↕	↕	↕	↕	↕
	ADDD	C3	4	3	D3	6	2	E3	6 +	2 +	F3	7	3				D + M:M + 1 → D	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	4	2	A4	4 +	2 +	B4	5	3				A ∧ M → A	•	↕	↕	0	•
	ANDB	C4	2	2	D4	4	2	E4	4 +	2 +	F4	5	3				B ∧ M → B	•	↕	↕	0	•
	ANDCC	1C	3	2													CC ∧ IMM → CC					7
ASL	ASLA													48	2	1		8	↕	↕	↕	↕
	ASLB													58	2	1		8	↕	↕	↕	↕
	ASL				08	6	2	68	6 +	2 +	78	7	3					8	↕	↕	↕	↕
ASR	ASRA													47	2	1		8	↕	↕	•	↕
	ASRB													57	2	1		8	↕	↕	•	↕
	ASR				07	6	2	67	6 +	2 +	77	7	3					8	↕	↕	•	↕
BIT	BITA	85	2	2	95	4	2	A5	4 +	2 +	B5	5	3				Bit Test A (M ∧ A)	•	↕	↕	0	•
	BITB	C5	2	2	D5	4	2	E5	4 +	2 +	F5	5	3				Bit Test B (M ∧ B)	•	↕	↕	0	•
CLR	CLRA													4F	2	1	0 → A	•	0	1	0	0
	CLRB													5F	2	1	0 → B	•	0	1	0	0
	CLR				0F	6	2	6F	6 +	2 +	7F	7	3				0 → M	•	0	1	0	0
CMP	CMPA	81	2	2	91	4	2	A1	4 +	2 +	B1	5	3				Compare M from A	↕	↕	↕	↕	↕
	CMPB	C1	2	2	D1	4	2	E1	4 +	2 +	F1	5	3				Compare M from B	↕	↕	↕	↕	↕
	CMPD	10	5	4	10	7	3	10	7 +	3 +	10	8	4				Compare M:M + 1 from D	•	↕	↕	↕	↕
		83			93			A3			B3											
	CMPS	11	5	4	11	7	3	11	7 +	3 +	11	8	4				Compare M:M + 1 from S	•	↕	↕	↕	↕
		8C			9C			AC			BC											
	CMPU	11	5	4	11	7	3	11	7 +	3 +	11	8	4				Compare M:M + 1 from U	•	↕	↕	↕	↕
		83			93			A3			B3											
	CMPX	8C	4	3	9C	6	2	AC	6 +	2 +	BC	7	3				Compare M:M + 1 from X	•	↕	↕	↕	↕
	CMPY	10	5	4	10	7	3	10	7 +	3 +	10	8	4				Compare M:M + 1 from Y	•	↕	↕	↕	↕
		8C			9C			AC			BC											
COM	COMA													43	2	1	$\overline{A} \rightarrow A$	•	↕	↕	0	1
	COMB													53	2	1	$\overline{B} \rightarrow B$	•	↕	↕	0	1
	COM				03	6	2	63	6 +	2 +	73	7	3				$\overline{M} \rightarrow M$	•	↕	↕	0	1
CWAI		3C	≥20	2													CC ∧ IMM → CC Wait for Interrupt					7
DAA														19	2	1	Decimal Adjust A	•	↕	↕	0	↕
DEC	DECA													4A	2	1	A − 1 → A	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	↕	↕	↕	•
	DEC				0A	6	2	6A	6 +	2 +	7A	7	3				M − 1 → M	•	↕	↕	↕	•
EOR	EORA	88	2	2	98	4	2	A8	4 +	2 +	B8	5	3				A ⊻ M → A	•	↕	↕	0	•
	EORB	C8	2	2	D8	4	2	E8	4 +	2 +	F8	5	3				B ⊻ M → B	•	↕	↕	0	•
EXG	R1, R2	1E	8	2													R1 ↔ R2[2]	•	•	•	•	•
INC	INCA													4C	2	1	A + 1 → A	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	↕	↕	↕	•
	INC				0C	6	2	6C	6 +	2 +	7C	7	3				M + 1 → M	•	↕	↕	↕	•
JMP					0E	3	2	6E	3 +	2 +	7E	4	3				EA[3] → PC	•	•	•	•	•
JSR					9D	7	2	AD	7 +	2 +	BD	8	3				Jump to Subroutine	•	•	•	•	•
LD	LDA	86	2	2	96	4	2	A6	4 +	2 +	B6	5	3				M → A	•	↕	↕	0	•
	LDB	C6	2	2	D6	4	2	E6	4 +	2 +	F6	5	3				M → B	•	↕	↕	0	•
	LDD	CC	3	3	DC	5	2	EC	5 +	2 +	FC	6	3				M:M + 1 → D	•	↕	↕	0	•
	LDS	10	4	4	10	6	3	10	6 +	3 +	10	7	4				M:M + 1 → S	•	↕	↕	0	•
		CE			DE			EE			FE											
	LDU	CE	3	3	DE	5	2	EE	5 +	2 +	FE	6	3				M:M + 1 → U	•	↕	↕	0	•
	LDX	8E	3	3	9E	5	2	AE	5 +	2 +	BE	6	3				M:M + 1 → X	•	↕	↕	0	•
	LDY	10	4	4	10	6	3	10	6 +	3 +	10	7	4				M:M + 1 → Y	•	↕	↕	0	•
		8E			9E			AE			BE											
LEA	LEAS							32	4 +	2 +							EA[3] → S	•	•	•	•	•
	LEAU							33	4 +	2 +							EA[3] → U	•	•	•	•	•
	LEAX							30	4 +	2 +							EA[3] → X	•	•	↕	•	•
	LEAY							31	4 +	2 +							EA[3] → Y	•	•	↕	•	•

LEGEND:

OP	Operation Code (Hexadecimal)	$\overline{M}$	Complement of M	↕	Test and set if true, cleared otherwise
~	Number of MPU Cycles	→	Transfer Into	•	Not Affected
#	Number of Program Bytes	H	Half-carry (from bit 3)	CC	Condition Code Register
+	Arithmetic Plus	N	Negative (sign bit)	:	Concatenation
−	Arithmetic Minus	Z	Zero result	V	Logical or
•	Multiply	V	Overflow, 2's complement	∧	Logical and
		C	Carry from ALU	⊻	Logical Exclusive or

FIGURE 18 — PROGRAMMING AID (CONTINUED)

Instruction	Forms	Addressing Modes: Immediate Op	Immediate ~	Immediate #	Direct Op	Direct ~	Direct #	Indexed[1] Op	Indexed[1] ~	Indexed[1] #	Extended Op	Extended ~	Extended #	Inherent Op	Inherent ~	Inherent #	Description	5 H	3 N	2 Z	1 V	0 C
LSL	LSLA													48	2	1	A ← 0	•	↕	↕	↕	↕
	LSLB													58	2	1	B ← 0	•	↕	↕	↕	↕
	LSL				08	6	2	68	6+	2+	78	7	3				M ← 0 (C, b7 … b0)	•	↕	↕	↕	↕
LSR	LSRA													44	2	1	A → 0	•	0	↕	•	↕
	LSRB													54	2	1	B → 0	•	0	↕	•	↕
	LSR				04	6	2	64	6+	2+	74	7	3				M → 0 (b7 … b0, C)	•	0	↕	•	↕
MUL														3D	11	1	A × B → D (Unsigned)	•	•	↕	•	9
NEG	NEGA													40	2	1	$\overline{A}$ + 1 → A	8	↕	↕	↕	↕
	NEGB													50	2	1	$\overline{B}$ + 1 → B	8	↕	↕	↕	↕
	NEG				00	6	2	60	6+	2+	70	7	3				$\overline{M}$ + 1 → M	8	↕	↕	↕	↕
NOP														12	2	1	No Operation	•	•	•	•	•
OR	ORA	8A	2	2	9A	4	2	AA	4+	2+	BA	5	3				A ∨ M → A	•	↕	↕	0	•
	ORB	CA	2	2	DA	4	2	EA	4+	2+	FA	5	3				B ∨ M → B	•	↕	↕	0	•
	ORCC	1A	3	2													CC ∨ IMM → CC				7	
PSH	PSHS	34	5+[4]	2													Push Registers on S Stack	•	•	•	•	•
	PSHU	36	5+[4]	2													Push Registers on U Stack	•	•	•	•	•
PUL	PULS	35	5+[4]	2													Pull Registers from S Stack	•	•	•	•	•
	PULU	37	5+[4]	2													Pull Registers from U Stack	•	•	•	•	•
ROL	ROLA													49	2	1	A ← C, b7 … b0	•	↕	↕	↕	↕
	ROLB													59	2	1	B ← C, b7 … b0	•	↕	↕	↕	↕
	ROL				09	6	2	69	6+	2+	79	7	3				M ← C, b7 … b0	•	↕	↕	↕	↕
ROR	RORA													46	2	1	A → C, b7 … b0	•	↕	↕	•	↕
	RORB													56	2	1	B → C, b7 … b0	•	↕	↕	•	↕
	ROR				06	6	2	66	6+	2+	76	7	3				M → C, b7 … b0	•	↕	↕	•	↕
RTI														3B	6/15	1	Return From Interrupt					7
RTS														39	5	1	Return from Subroutine	•	•	•	•	•
SBC	SBCA	82	2	2	92	4	2	A2	4+	2+	B2	5	3				A - M - C → A	8	↕	↕	↕	↕
	SBCB	C2	2	2	D2	4	2	E2	4+	2+	F2	5	3				B - M - C → B	8	↕	↕	↕	↕
SEX														1D	2	1	Sign Extend B into A	•	↕	↕	0	•
ST	STA				97	4	2	A7	4+	2+	B7	5	3				A → M	•	↕	↕	0	•
	STB				D7	4	2	E7	4+	2+	F7	5	3				B → M	•	↕	↕	0	•
	STD				DD	5	2	ED	5+	2+	FD	6	3				D → M:M + 1	•	↕	↕	0	•
	STS				10	6	3	10	6+	3+	10	7	4				S → M:M + 1	•	↕	↕	0	•
					DF			EF			FF											
	STU				DF	5	2	EF	5+	2+	FF	6	3				U → M:M + 1	•	↕	↕	0	•
	STX				9F	5	2	AF	5+	2+	BF	6	3				X → M:M + 1	•	↕	↕	0	•
	STY				10	6	3	10			10	7	4				Y → M:M + 1	•	↕	↕	0	•
					9F			AF	6+	3+	BF											
SUB	SUBA	80	2	2	90	4	2	A0	4+	2+	B0	5	3				A - M → A	8	↕	↕	↕	↕
	SUBB	C0	2	2	D0	4	2	E0	4+	2+	F0	5	3				B - M → B	8	↕	↕	↕	↕
	SUBD	83	4	3	93	6	2	A3	6+	2+	B3	7	3				D - M:M + 1 → D	•	↕	↕	↕	↕
SWI	SWI[6]													3F	19	1	Software Interrupt 1	•	•	•	•	•
	SWI2[6]													10	20	2	Software Interrupt 2	•	•	•	•	•
														3F								
	SWI3[6]													11	20	1	Software Interrupt 3	•	•	•	•	•
														3F								
SYNC														13	≥ 4	1	Synchronize to Interrupt	•	•	•	•	•
TFR	R1, R2	1F	6	2													R1 → R2[2]	•	•	•	•	•
TST	TSTA													4D	2	1	Test A	•	↕	↕	0	•
	TSTB													5D	2	1	Test B	•	↕	↕	0	•
	TST				0D	6	2	6D	6+	2+	7D	7	3				Test M	•	↕	↕	0	•

NOTES:

1. This column gives a base cycle and byte count. To obtain total count, add the values obtained from the INDEXED ADDRESSING MODE table, Table 2.
2. R1 and R2 may be any pair of 8 bit or any pair of 16 bit registers
   The 8 bit registers are: A, B, CC, DP
   The 16 bit registers are: X, Y, U, S, D, PC
3. EA is the effective address.
4. The PSH and PUL instructions require 5 cycles plus 1 cycle for each **byte** pushed or pulled.
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken (Branch instructions).
6. SWI sets I and F bits. SWI2 and SWI3 do not affect I and F.
7. Conditions Codes set as a direct result of the instruction.
8. Vaue of half-carry flag is undefined.
9. Special Case — Carry set if b7 is SET.

FIGURE 18 — PROGRAMMING AID (CONTINUED)

Branch Instructions

Instruction	Forms	Addressing Mode Relative OP	~	#	Description	5 H	3 N	2 Z	1 V	0 C
BCC	BCC	24	3	2	Branch C = 0	•	•	•	•	•
	LBCC	10	5(6)	4	Long Branch C = 0	•	•	•	•	•
		24								
BCS	BCS	25	3	2	Branch C = 1	•	•	•	•	•
	LBCS	10	5(6)	4	Long Branch C = 1	•	•	•	•	•
		25								
BEQ	BEQ	27	3	2	Branch Z = 1	•	•	•	•	•
	LBEQ	10	5(6)	4	Long Branch Z = 1	•	•	•	•	•
		27								
BGE	BGE	2C	3	2	Branch ≥ Zero	•	•	•	•	•
	LBGE	10	5(6)	4	Long Branch ≥ Zero	•	•	•	•	•
		2C								
BGT	BGT	2E	3	2	Branch > Zero	•	•	•	•	•
	LBGT	10	5(6)	4	Long Branch > Zero	•	•	•	•	•
		2E								
BHI	BHI	22	3	2	Branch Higher	•	•	•	•	•
	LBHI	10	5(6)	4	Long Branch Higher	•	•	•	•	•
		22								
BHS	BHS	24	3	2	Branch Higher or Same	•	•	•	•	•
	LBHS	10	5(6)	4	Long Branch Higher or Same	•	•	•	•	•
		24								
BLE	BLE	2F	3	2	Branch ≤ Zero	•	•	•	•	•
	LBLE	10	5(6)	4	Long Branch ≤ Zero	•	•	•	•	•
		2F								
BLO	BLO	25	3	2	Branch lower	•	•	•	•	•
	LBLO	10	5(6)	4	Long Branch Lower	•	•	•	•	•
		25								

Instruction	Forms	Addressing Mode Relative OP	~	#	Description	5 H	3 N	2 Z	1 V	0 C
BLS	BLS	23	3	2	Branch Lower or Same	•	•	•	•	•
	LBLS	10	5(6)	4	Long Branch Lower or Same	•	•	•	•	•
		23								
BLT	BLT	2D	3	2	Branch < Zero	•	•	•	•	•
	LBLT	10	5(6)	4	Long Branch < Zero	•	•	•	•	•
		2D								
BMI	BMI	2B	3	2	Branch Minus	•	•	•	•	•
	LBMI	10	5(6)	4	Long Branch Minus	•	•	•	•	•
		2B								
BNE	BNE	26	3	2	Branch Z = 0	•	•	•	•	•
	LBNE	10	5(6)	4	Long Branch Z = 0	•	•	•	•	•
		26								
BPL	BPL	2A	3	2	Branch Plus	•	•	•	•	•
	LBPL	10	5(6)	4	Long Branch Plus	•	•	•	•	•
		2A								
BRA	BRA	20	3	2	Branch Always	•	•	•	•	•
	LBRA	16	5	3	Long Branch Always	•	•	•	•	•
BRN	BRN	21	3	2	Branch Never	•	•	•	•	•
	LBRN	10	5	4	Long Branch Never	•	•	•	•	•
		21								
BSR	BSR	8D	7	2	Branch to Subroutine	•	•	•	•	•
	LBSR	17	9	3	Long Branch to Subroutine	•	•	•	•	•
BVC	BVC	28	3	2	Branch V = 0	•	•	•	•	•
	LBVC	10	5(6)	4	Long Branch V = 0	•	•	•	•	•
		28								
BVS	BVS	29	3	2	Branch V = 1	•	•	•	•	•
	LBVS	10	5(6)	4	Long Branch V = 1	•	•	•	•	•
		29								

SIMPLE BRANCHES

	OP	~	#
BRA	20	3	2
LBRA	16	5	3
BRN	21	3	2
LBRN	1021	5	4
BSR	8D	7	2
LBSR	17	9	3

SIMPLE CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
N = 1	BMI	2B	BPL	2A
Z = 1	BEQ	27	BNE	26
V = 1	BVS	29	BVC	28
C = 1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BGT	2E	BLE	2F
r ≥ m	BGE	2C	BLT	2D
r = m	BEQ	27	BNE	26
r ≤ m	BLE	2F	BGT	2E
r < m	BLT	2D	BGE	2C

UNSIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BHI	22	BLS	23
r ≥ m	BHS	24	BLO	25
r = m	BEQ	27	BNE	26
r ≤ m	BLS	23	BHI	22
r < m	BLO	25	BHS	24

NOTES:

1. All conditional branches have both short and long variations.
2. All short branches are 2 bytes and require 3 cycles.
3. All conditional long branches are formed by prefixing the short branch opcode with $10 and using a 16-bit destination offset.
4. All conditional long branches require 4 bytes and 6 cycles if the branch is taken or 5 cycles if the branch is not taken.
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken.

INDEXED ADDRESSING MODES

Type	Forms	Nondirect: Assembler Form	Nondirect: Post-Byte Opcode	Nondirect: + ~	Nondirect: + #	Indirect: Assembler Form	Indirect: Post-Byte Opcode	Indirect: + ~	Indirect: + #
Constant Offset From R	No Offset	, R	1RR00100	0	0	[, R]	1RR10100	3	0
	5-Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8-Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16-Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R	A – Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
	B – Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D – Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	, R+	1RR00000	2	0	not allowed			
	Increment By 2	, R + +	1RR00001	3	0	[, R + +]	1RR10001	6	0
	Decrement By 1	, -R	1RR00010	2	0	not allowed			
	Decrement By 2	, --R	1RR00011	3	0	[, --R]	1RR10011	6	0
Constant Offset From PC	8-Bit Offset	n, PCR	1XX01100	1	1	[n, PCR]	1XX11100	4	1
	16-Bit Offset	n, PCR	1XX01101	5	2	[n, PCR]	1XX11101	8	2
Extended Indirect	16-Bit Address	–	–	–	–	[n]	10011111	5	2

R = X, Y, U, or S
X = Don't Care

RR: 00 = X 10 = U
01 = Y 11 = S

INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

Post-Byte Register Bit 7	6	5	4	3	2	1	0	Indexed Addressing Mode
0	R	R	x	x	x	x	x	EA = , R + 5 Bit Offset
1	R	R	0	0	0	0	0	, R +
1	R	R	I	0	0	0	1	, R + +
1	R	R	0	0	0	1	0	,- R
1	R	R	I	0	0	1	1	, -- R
1	R	R	I	0	1	0	0	EA = , R + 0 Offset
1	R	R	I	0	1	0	1	EA = , R + ACCB Offset
1	R	R	I	0	1	1	0	EA = , R + ACCA Offset
1	R	R	I	1	0	0	0	EA = , R+ 8-Bit Offset
1	R	R	I	1	0	0	1	EA = , R + 16-Bit Offset
1	R	R	I	1	0	1	1	EA = , R + D Offset
1	x	x	I	1	1	0	0	EA = , PC + 8-Bit Offset
1	x	x	I	1	1	0	1	EA = , PC + 16-Bit Offset
1	R	R	1	1	1	1	1	EA = [, Address]

Addressing Mode Field

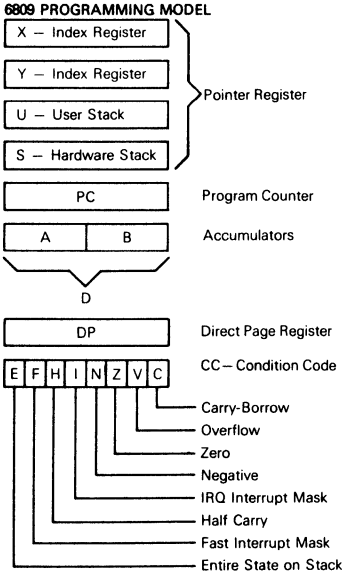
Indirect Field

(Sign bit when b7 = 0)

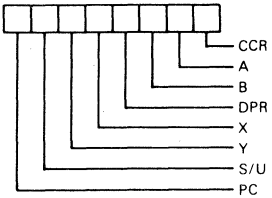
Register Field: RR
00 = X
01 = Y
10 = U
11 = S

X = Don't Care

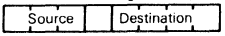
6809 PROGRAMMING MODEL



Push/Pull Post Byte



Transfer/Exchange Post Byte



Register Field

0000 = D (A-B)	0101 = PC
0001 = X	1000 = A
0010 = Y	1001 = B
0011 = U	1010 = CCR
0100 = S	1011 = DPR

6809 Stacking Order

6809 Vectors

FFFE	Restart
FFFC	NMI
FFFA	SWI
FFF8	IRQ
FFF6	FIRQ
FFF4	SW12
FFF2	SW13
FFF0	Reserved

ORDERING INFORMATION

EF68A09E C M B/B

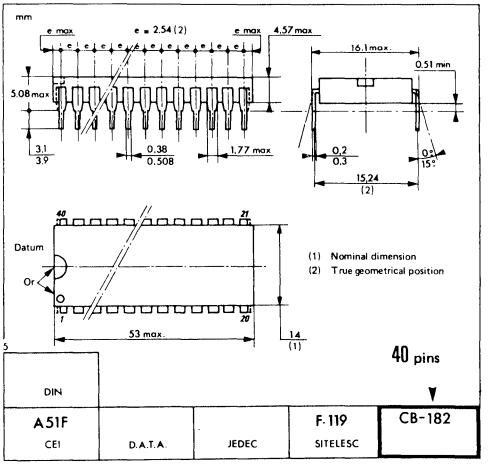
Device — Package — Oper. temp. — Screening level

The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

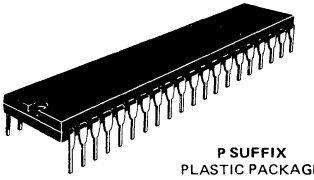
DEVICE	PACKAGE					OPER. TEMP			SCREENING LEVEL			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6809E (1.0 MHz)	●	●	●		●	●			●			
	●	●					●		●			
	●							●	●		●	●
		●						●	●		●	
EF68A09E (1.5 MHz)	●	●	●			●			●			
	●	●					●		●			
	●							●	●		●	●
		●						●	●		●	
EF68B09E (2.0 MHz)	●	●	●			●			●			
	●	●					●		●		●	

Examples : EF6809EC, EF6809ECV, EF6809ECM

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** −40°C to +85°C, **M :** −55°C to +125°C, *** :** may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.



CB-182

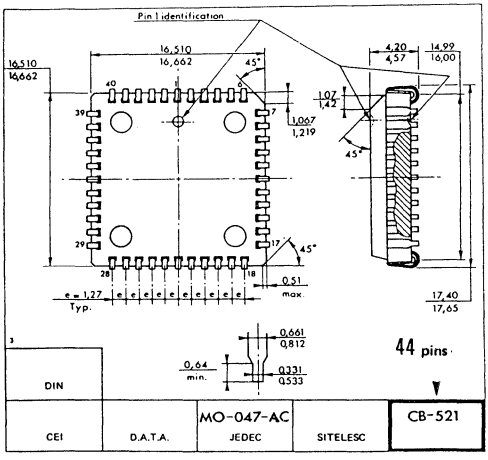


P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

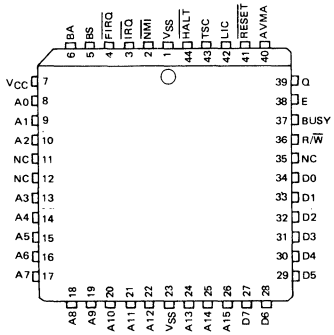
J SUFFIX
CERDIP PACKAGE

C SUFFIX
CERAMIC PACKAGE



CB-521

FN SUFFIX
PLCC 44



CHAPTER 2 - 6800 PERIPHERALS

6800 PERIPHERALS SELECTION GUIDE

Part number	Description	Technology	Alt source	CLK freq. (MHz)	Page
EF6821 EF68A21 EF68B21	Peripheral Interface Adapter (PIA) Two bidirectional 8-bit buses for interface to peripherals - Two programmable control registers - Two programmable data direction registers	NMOS	MC6821 MC68A21 MC68B21	1 1.5 2	2-3
EF6840 EF68A40 EF68B40	Programmable Timer Module (PTM) Three 16-bit - Binary counters - Selectable gating For frequency or pulse-width comparison	NMOS	MC6840 MC68A40 MC68B50	1 1.5 2	2-15
EF6850 EF68A50 EF68B50	Asynchronous Communication Interface Adapter (ACIA) - 8 and 9-bit transmission - Peripheral/modem control functions	NMOS	MC6850 MC68A50 MC68B50	1 1.5 2	2-31
EF6854 EF68A54 EF68B54	Advanced Data-Link Controller (ADLC)	NMOS	MC6854 MC68A54 MC68B54	1 1.5 2	2-41

THOMSON COMPONENTS MOSTEK

EF6821

PERIPHERAL INTERFACE ADAPTER (PIA)

The EF6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the 6800 family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines; Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- High-Impedance Three-State and Direct Transistor Drive Peripheral Lines
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on Side A Peripheral Lines
- Two TTL Drive Capability on All A and B Side Buffers
- TTL-Compatible
- Static Operation
- Three available versions : EF6821 (1.0 MHz)
  EF68A21 (1.5 MHz)
  EF68B21 (2.0 MHz)

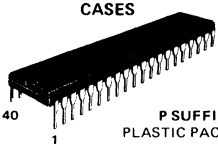
MOS

(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

PERIPHERAL INTERFACE ADAPTER

CASES

CB-182



P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

J SUFFIX
CERDIP PACKAGE

C SUFFIX
CERAMIC PACKAGE

CB-521

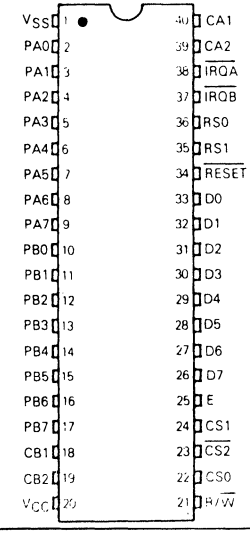
FN SUFFIX
PLCC 44

CB-708

E SUFFIX
LCCC 44

Hi-Rel versions available - See chapter 9

PIN ASSIGNMENT



MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range EF6821, EF68A21, EF68B21 EF6821, EF68A21, EF68B21 : V suffix EF6821, EF68A21 : M suffix	T_A	T_L to T_H 0 to 70 −40 to +85 −55 to +125	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $GND \leq (V_{in}$ or $V_{out}) \leq V_{CC}$.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either GND or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Plastic Cerdip PLCC	θ_{JA}	 50 100 60 100	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K + (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
BUS CONTROL INPUTS (R/$\overline{W}$, Enable, $\overline{RESET}$, RS0, RS1, CS0, CS1, CS2)					
Input High Voltage	V_{IH}	V_{SS} + 2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	I_{in}	–	1.0	2.5	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	7.5	pF
INTERRUPT OUTPUTS ($\overline{IRQA}$, $\overline{IRQB}$)					
Output Low Voltage (I_{Load} = 1.6 mA)	V_{OL}	–	–	V_{SS} + 0.4	V
Hi-Z Output Leakage Current	I_{OZ}	–	1.0	10	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{out}	–	–	5.0	pF
DATA BUS (D0-D7)					
Input High Voltage	V_{IH}	V_{SS} + 2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
Hi-Z Input Leakage Current (V_{in} = 0.4 to 2.4 V)	I_{IZ}	–	2.0	10	μA
Output High Voltage (I_{Load} = − 205 μA)	V_{OH}	V_{SS} + 2.4	–	–	V
Output Low Voltage (I_{Load} = 1.6 mA)	V_{OL}	–	–	V_{SS} + 0.4	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	12.5	pF

DC ELECTRICAL CHARACTERISTICS (Continued)

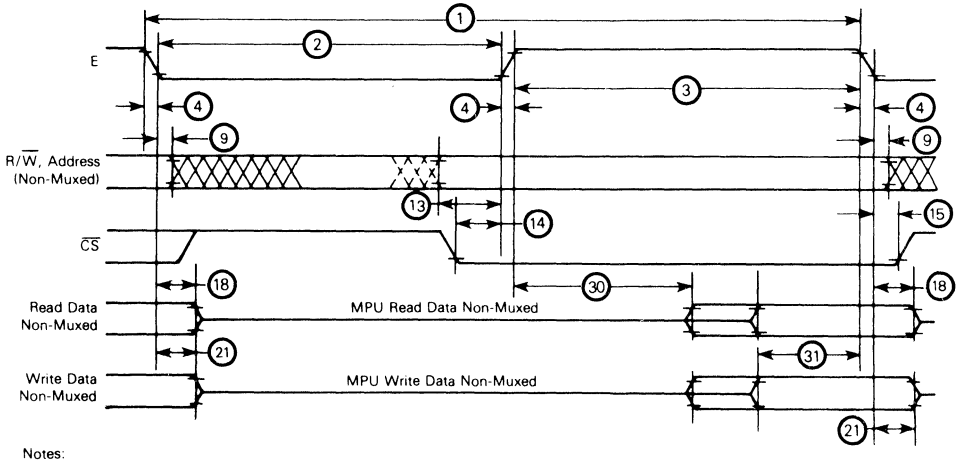
Characteristic		Symbol	Min	Typ	Max	Unit
PERIPHERAL BUS (PA0-PA7, PB0-PB7, CA1, CA2, CB1, CB2)						
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/$\overline{W}$, $\overline{RESET}$, RS0, RS1, CS0, CS1, $\overline{CS2}$, CA1, CB1, Enable	I_{in}	–	1.0	2.5	μA
Hi-Z Input Leakage Current (V_{in} = 0.4 to 2.4 V)	PB0-PB7, CB2	I_{IZ}	–	2.0	10	μA
Input High Current (V_{IH} = 2.4 V)	PA0-PA7, CA2	I_{IH}	– 200	– 400	–	μA
Darlington Drive Current (V_O = 1.5 V)	PB0-PB7, CB2	I_{OH}	– 1.0	–	– 10	mA
Input Low Current (V_{IL} = 0.4 V)	PA0-PA7, CA2	I_{IL}	–	– 1.3	– 2.4	mA
Output High Voltage (I_{Load} = – 200 μA) (I_{Load} = – 10 μA)	 PA0-PA7, PB0-PB7, CA2, CB2 PA0-PA7, CA2	V_{OH}	 V_{SS} + 2.4 V_{CC} – 1.0	 – –	 – –	V
Output Low Voltage (I_{Load} = 3.2 mA)		V_{OL}	–	–	V_{SS} + 0.4	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)		C_{in}	–	–	10	pF
POWER REQUIREMENTS						
Internal Power Dissipation (Measured at T_L = 0°C)		P_{INT}	–	–	550	mW

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	EF6821		EF68A21		EF68B21		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

FIGURE 1 — BUS TIMING



Notes:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

PERIPHERAL TIMING CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0 V, $T_A = T_L$ to T_H unless otherwise specified)

Characteristic	Symbol	EF6821		EF68A21		EF68B21		Unit	Reference Fig. No.
		Min	Max	Min	Max	Min	Max		
Data Setup Time	t_{PDS}	200	–	135	–	100	–	ns	6
Data Hold Time	t_{PDH}	0	–	0	–	0	–	ns	6
Delay Time, Enable Negative Transition to CA2 Negative Transition	t_{CA2}	–	1.0	–	0.670	–	0.500	µs	3, 7, 8
Delay Time, Enable Negative Transition to CA2 Positive Transition	T_{RS1}	–	1.0	–	0.670	–	0.500	µs	3, 7
Rise and Fall Times for CA1 and CA2 Input Signals	t_r, t_f	–	1.0	–	1.0	–	1.0	µs	8
Delay Time from CA1 Active Transition to CA2 Positive Transition	t_{RS2}	–	2.0	–	1.35	–	1.0	µs	3, 8
Delay Time, Enable Negative Transition to Data Valid	t_{PDW}	–	1.0	–	0.670	–	0.5	µs	3, 9, 10
Delay Time, Enable Negative Transition to CMOS Data Valid PA0-PA7, CA2	t_{CMOS}	–	2.0	–	1.35	–	1.0	µs	4, 9
Delay Time, Enable Positive Transition to CB2 Negative Transition	t_{CB2}	–	1.0	–	0.670	–	0.5	µs	3, 11, 12
Delay Time, Data Valid to CB2 Negative Transition	t_{DC}	20	–	20	–	20	–	ns	3, 10
Delay Time, Enable Positive Transition to CB2 Positive Transition	t_{RS1}	–	1.0	–	0.670	–	0.5	µs	3, 11
Control Output Pulse Width, CA2/CB2	PW_{CT}	500	–	375	–	250	–	ns	3, 11
Rise and Fall Time for CB1 and CB2 Input Signals	t_r, t_f	–	1.0	–	1.0	–	1.0	µ	12
Delay Time, CB1 Active Transition to CB2 Positive Transition	t_{RS2}	–	2.0	–	1.35	–	1.0	µs	3, 12
Interrupt Release Time, $\overline{IRQA}$ and $\overline{IRQB}$	t_{IR}	–	1.60	–	1.10	–	0.85	µs	5, 14
Interrupt Response Time	t_{RS3}	–	1.0	–	1.0	–	1.0	µs	5, 13
Interrupt Input Pulse Time	PW_I	500	–	500	–	500	–	ns	13
$\overline{RESET}$ Low Time*	t_{RL}	1.0	–	0.66	–	0.5	–	µs	15

*The $\overline{RESET}$ line must be high a minimum of 1.0 µs before addressing the PIA.

FIGURE 2 – BUS TIMING TEST LOADS

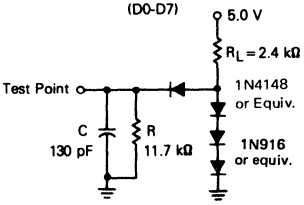


FIGURE 3 – TTL EQUIVALENT TEST LOAD

(PA0–PA7, PB0–PB7, CA2, CB2)

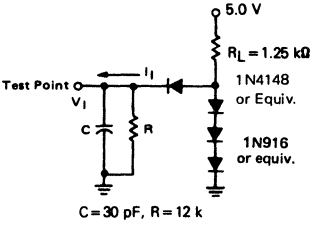


FIGURE 4 – CMOS EQUIVALENT TEST LOAD

(PA0-PA7, CA2)

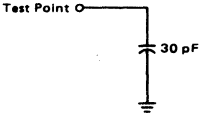


FIGURE 5 – NMOS EQUIVALENT TEST LOAD

($\overline{IRQ}$ Only)

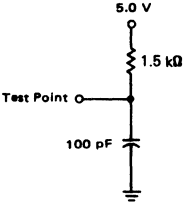


FIGURE 6 — PERIPHERAL DATA SETUP AND HOLD TIMES
(Read Mode)

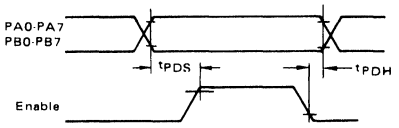
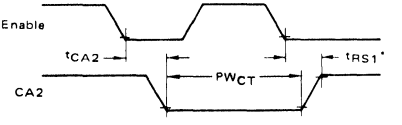


FIGURE 7 — CA2 DELAY TIME
(Read Mode; CRA-5 = CRA3 = 1, CRA-4 = 0)



*Assumes part was deselected during the previous E pulse.

FIGURE 8 — CA2 DELAY TIME
(Read Mode; CRA-5 = 1, CRA-3 = CRA-4 = 0)

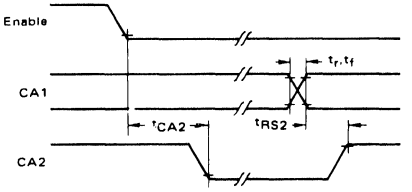


FIGURE 9 — PERIPHERAL CMOS DATA DELAY TIMES
(Write Mode; CRA-5 = CRA-3 = 1, CRA-4 = 0)

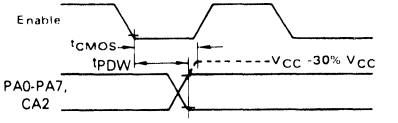
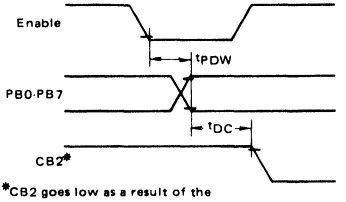
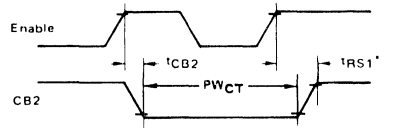


FIGURE 10 — PERIPHERAL DATA AND CB2 DELAY TIMES
(Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)



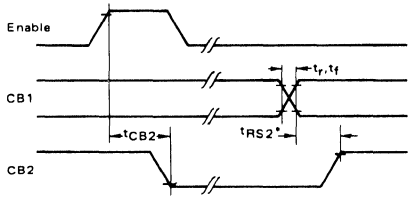
*CB2 goes low as a result of the positive transition of Enable.

FIGURE 11 — CB2 DELAY TIME
(Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)



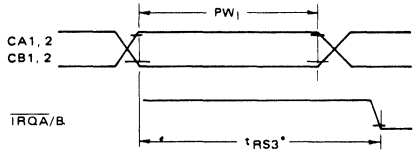
*Assumes part was deselected during the previous E pulse

FIGURE 12 — CB2 DELAY TIME
(Write Mode; CRB-5 = 1, CRB-3 = CRB-4 = 0)



*Assumes part was deselected during any previous E pulse.

FIGURE 13 — INTERRUPT PULSE WIDTH AND $\overline{IRQ}$ RESPONSE



*Assumes Interrupt Enable Bits are set.

Note: Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 14 — $\overline{IRQ}$ RELEASE TIME

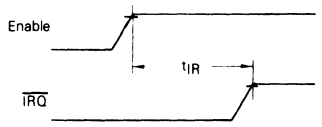
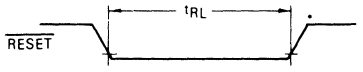


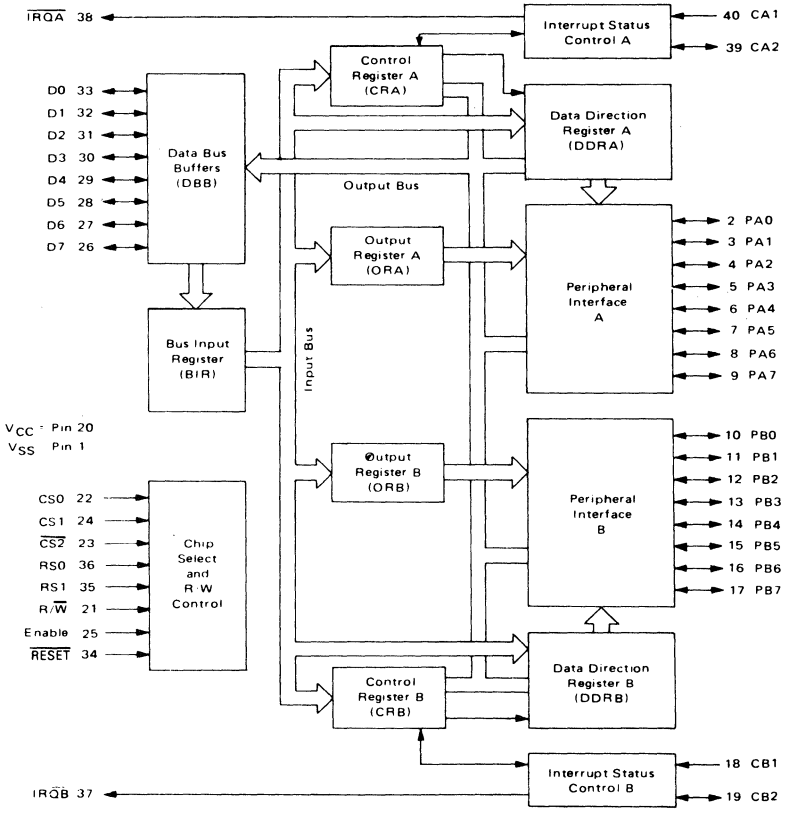
FIGURE 15 — $\overline{RESET}$ LOW TIME



*The $\overline{RESET}$ line must be a V_{IH} for a minimum of 1.0 μs before addressing the PIA.

Note: Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 16 — EXPANDED BLOCK DIAGRAM



PIA INTERFACE SIGNALS FOR MPU

The PIA interfaces to the 6800 bus with an 8-bit bidirectional data bus, three chip select lines, two register select lines, two interrupt request lines, a read/write line, an enable line and a reset line. To ensure proper operation with the EF6800, EF6802, or EF6808 microprocessors, VMA should be used as an active part of the address decoding.

Bidirectional Data (D0-D7) – The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the PIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a PIA read operation. The read/write line is in the read (high) state when the PIA is selected for a read operation.

Enable (E) – The enable pulse, E, is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse.

Read/Write (R/$\overline{W}$) – This signal is generated by the MPU to control the direction of data transfers on the data bus. A low state on the PIA read/write line enables the input buffers and data is transferred from the MPU to the PIA on the E signal if the device has been selected. A high on the read/write line sets up the PIA for a transfer of data to the bus. The PIA output buffers are enabled when the proper address and the enable pulse E are present.

$\overline{RESET}$ – The active low $\overline{RESET}$ line is used to reset all register bits in the PIA to a logical zero (low). This line can be used as a power-on reset and as a master reset during system operation.

Chip Selects (CS0, CS1, and $\overline{CS2}$) – These three input signals are used to select the PIA. CS0 and CS1 must be high and $\overline{CS2}$ must be low for selection of the device. Data transfers are then performed under the control of the enable and read/write signals. The chip select lines must be stable for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

Register Selects (RS0 and RS1) – The two register select lines are used to select the various registers inside the PIA. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

Interrupt Request ($\overline{IRQA}$ and $\overline{IRQB}$) – The active low Interrupt Request lines ($\overline{IRQA}$ and $\overline{IRQB}$) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load device on the chip). This permits all interrupt request lines to be tied together in a wire-OR configuration.

Each Interrupt Request line has two internal interrupt flag bits that can cause the Interrupt Request line to go low. Each flag bit is associated with a particular peripheral interrupt line. Also, four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Servicing an interrupt by the MPU may be accomplished by a software routine that, on a prioritized basis, sequentially reads and tests the two control registers in each PIA for interrupt flag bits that are set.

The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the PIA is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA1, CA2, CB1, CB2). When these lines are used as interrupt inputs, at least one E pulse must occur from the inactive edge to the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8-bit bidirectional data buses and four interrupt/control lines for interfacing to peripheral devices.

Section A Peripheral Data (PA0-PA7) – Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines. In the input mode, the internal pullup resistor on these lines represents a maximum of 1.5 standard TTL loads.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written into the register will cause a "high" on the corresponding data line while a "0" results in a "low." Data in Output Register A may be read by an MPU "Read Peripheral Data A" operation when the corresponding lines are programmed as outputs. This data will be read properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

Section B Peripheral Data (PB0-PB7) – The peripheral data lines in the B Section of the PIA can be programmed to act as either inputs or outputs in a similar manner to PA0-PA7. They have three-state capability, allowing them to enter a high-impedance state when the peripheral data line is used as an input. In addition, data on the peripheral data lines

PB0-PB7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "high" or above 0.8 V for a "low". As outputs, these lines are compatible with standard TTL and may also be used as a source of at least 1 milliampere at 1.5 volts to directly drive the base of a transistor switch.

Interrupt Input (CA1 and CB1) – Peripheral input lines CA1 and CB1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

Peripheral Control (CA2) – The peripheral control line CA2 can be programmed to act as an interrupt input or as a peripheral control output. As an output, this line is compatible with standard TTL; as an input the internal pullup resistor on this line represents 1.5 standard TTL loads. The function of this signal line is programmed with Control Register A.

Peripheral Control (CB2) – Peripheral Control line CB2 may also be programmed to act as an interrupt input or peripheral control output. As an input, this line has high input impedance and is compatible with standard TTL. As an output it is compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor switch. This line is programmed by Control Register B.

INTERNAL CONTROLS

INITIALIZATION

A $\overline{\text{RESET}}$ has the effect of zeroing all PIA registers. This will set PA0-PA7, PB0-PB7, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS0 and RS1 inputs together with bit 2 in the Control Register, as shown in Table 1.

Details of possible configurations of the Data Direction and Control Register are as follows:

TABLE 1 – INTERNAL ADDRESSING

		Control Register Bit		
RS1	RS0	CRA-2	CRB-2	Location Selected
0	0	1	X	Peripheral Register A
0	0	0	X	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X Don't Care

PORT A-B HARDWARE CHARACTERISTICS

As shown in Figure 17, the EF6821 has a pair of I/O ports whose characteristics differ greatly. The A side is designed to drive CMOS logic to normal 30% to 70% levels, and incorporates an internal pullup device that remains connected even in the input mode. Because of this, the A side requires more drive current in the input mode than Port B. In contrast, the B side uses a normal three-state NMOS buffer which cannot pullup to CMOS levels without external resistors. The B side can drive extra loads such as Darlingtons without problem. When the PIA comes out of reset, the A port represents inputs with pullup resistors, whereas the B side (input mode also) will float high or low, depending upon the load connected to it.

Notice the differences between a Port A and Port B read operation when in the output mode. When reading Port A, the actual pin is read, whereas the B side read comes from an output latch, ahead of the actual pin.

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1, and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA1, CA2, CB1, or CB2. The format of the control words is shown in Figure 18.

DATA DIRECTION ACCESS CONTROL BIT (CRA-2 and CRB-2)

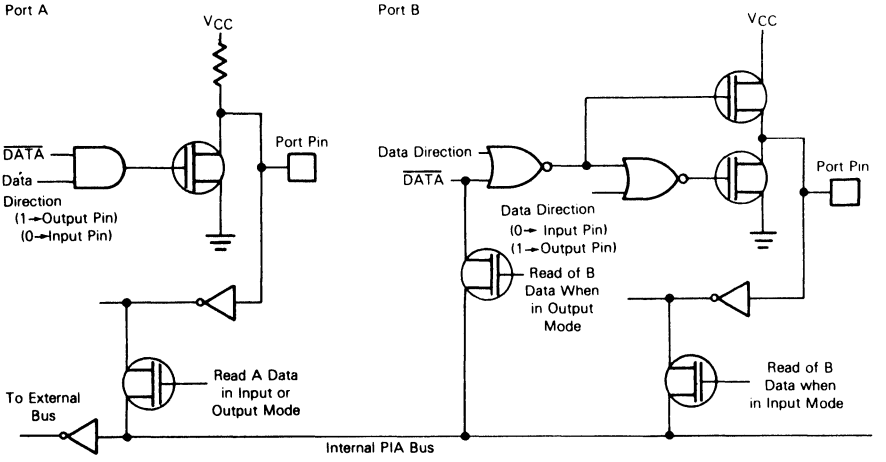
Bit 2, in each Control Register (CRA and CRB), determines selection of either a Peripheral Output Register or the corresponding Data Direction E Register when the proper register select signals are applied to RS0 and RS1. A "1" in bit 2 allows access of the Peripheral Interface Register, while a "0" causes the Data Direction Register to be addressed.

Interrupt Flags (CRA-6, CRA-7, CRB-6, and CRB-7) – The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4, and CRB-5) – Bits 3, 4, and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input line similar to CA1 (CB1). When CRA-5 (CRB-5) is high, CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different loading characteristics.

Control of CA1 and CB1 Interrupt Input Lines (CRA-0, CRB-0, CRA-1, and CRB-1) – The two lowest-order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to enable the MPU interrupt signals $\overline{IRQA}$ and $\overline{IRQB}$, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1.

FIGURE 17 – PORT A AND PORT B EQUIVALENT CIRCUITS



ORDERING INFORMATION

EF68A21 | C | M | B/B

- EF68A21: Device
- C: Package
- M: Oper. temp.
- B/B: Screening level

The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

DEVICE	PACKAGE					OPER. TEMP			SCREENING LEVEL			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6821 (1.0 MHz)	●	●	●		●	●			●			
	●	●	●				●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68A21 (1.5 MHz)	●	●	●			●			●			
	●	●	●				●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68B21 (2.0 MHz)	●	●	●			●			●			
	●	●					●		●		●	

Examples : EF6821C, EF6821CV, EF6821CM, EF6821JM

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** −40°C to +85°C, **M :** −55°C to +125°C, *** :** may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.

FIGURE 18 — CONTROL WORD FORMAT

Determine Active CA1 (CB1) Transition for Setting Interrupt Flag IRQA(B)1 — (bit 7)

b1 = 0: IRQA(B)1 set by high-to-low transition on CA1 (CB1)

b1 = 1: IRQA(B)1 set by low-to-high transition on CA1 (CB1).

IRQA(B) 1 Interrupt Flag (bit 7)

Goes high on active transition of CA1 (CB1); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

CA1 (CB1) Interrupt Request Enable/Disable

b0 = 0: Disables IRQA(B) MPU Interrupt by CA1 (CB1) active transition.[1]

b0 = 1: Enable IRQA(B) MPU Interrupt by CA1 (CB1) active transition.

1. IRQA(B) will occur on next (MPU generated) positive transition of b0 if CA1 (CB1) active transition occurred while interrupt was disabled.

Control Register

b7	b6	b5	b4	b3	b2	b1	b0
IRQA(B)1 Flag	IRQA(B)2 Flag	CA2 (CB2) Control			DDR Access	CA1 (CB1) Control	

IRQA(B)2 Interrupt Flag (bit 6)

When CA2 (CB2) is an input, IRQA(B) goes high on active transition CA2 (CB2); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

CA2 (CB2) Established as Output (b5 = 1): IRQA(B) 2 = 0, not affected by CA2 (CB2) transitions.

Determines Whether Data Direction Register Or Output Register is Addressed

b2 = 0: Data Direction Register selected.

b2 = 1: Output Register selected.

CA2 (CB2) Established as Output by b5 = 1

(Note that operation of CA2 and CB2 output functions are not identical)

b5 b4 b3

1 0 → CA2

b3 = 0: **Read Strobe with CA1 Restore.** CA2 goes low on first high-to-low E transition following an MPU read of Output Register A; returned high by next active CA1 transition, as specified by bit 1.

b3 = 1: **Read Strobe with E Restore** CA2 goes low on first high-to-low E transition following an MPU read of Output Register A; returned high by next high-to-low E transition during a deselect.

→ CB2

b3 = 0: **Write Strobe with CB1 Restore** CB2 goes low on first low-to-high E transition following an MPU write into Output Register B; returned high by the next active CB1 transition as specified by bit 1. CRB-b7 must first be cleared by a read of data.

b3 = 1: **Write Strobe with E Restore** CB2 goes low on first low-to-high E transition following an MPU write into Output Register B; returned high by the next low-to-high E transition following an E pulse which occurred while the part was deselected.

b5 b4 b3

1 1 → **Set/Reset CA2 (CB2)**

CA2 (CB2) goes low as MPU writes b3 = 0 into Control Register.

CA2 (CB2) goes high as MPU writes b3 = 1 into Control Register.

CA2 (CB2) Established as Input by b5 = 0

b5 b4 b3

0 → **CA2 (CB2) Interrupt Request Enable/Disable**

b3 = 0: Disables IRQA(B) MPU Interrupt by CA2 (CB2) active transition.*

b3 = 1: Enables IRQA(B) MPU Interrupt by CA2 (CB2) active transition.

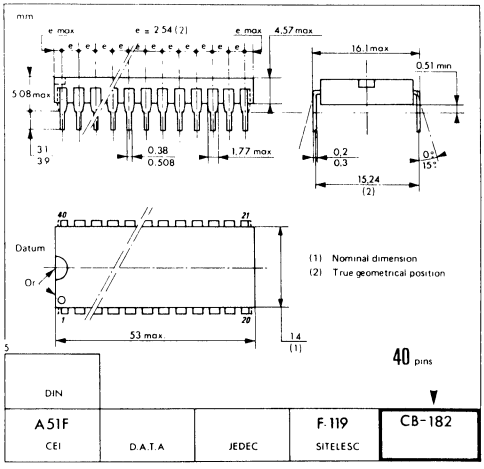
*IRQA(B) will occur on next (MPU generated) positive transition of b3 if CA2 (CB2) active transition occurred while interrupt was disabled.

→ **Determines Active CA2 (CB2) Transition for Setting Interrupt Flag IRQA(B)2 — (Bit b6)**

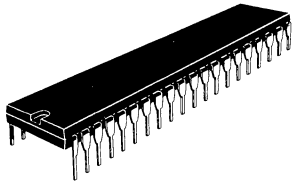
b4 = 0: IRQA(B)2 set by high-to-low transition on CA2 (CB2).

b4 = 1: IRQA(B)2 set by low-to-high transition on CA2 (CB2).

PHYSICAL DIMENSIONS



CB-182



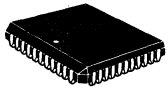
P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

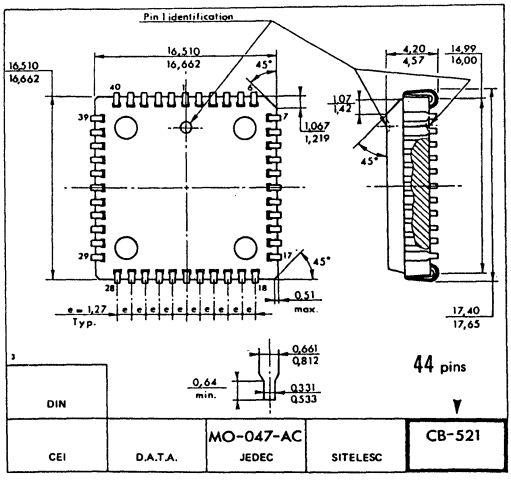
J SUFFIX
CERDIP PACKAGE

C SUFFIX
CERAMIC PACKAGE

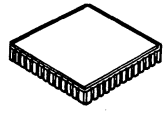
CB-521



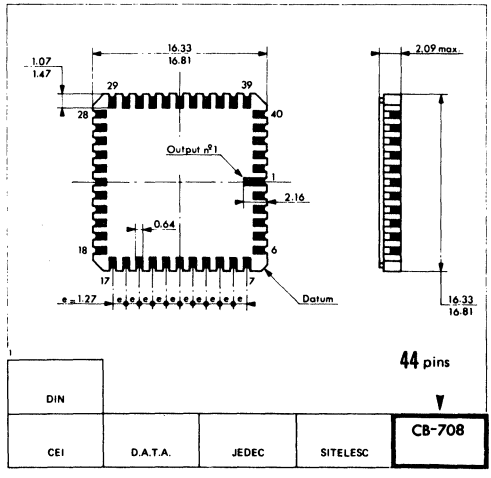
FN SUFFIX
PLCC 44



CB-708



E SUFFIX
LCCC 44



THOMSON COMPONENTS MOSTEK

EF6840

PROGRAMMABLE TIMER MODULE (PTM)

The EF6840 is a programmable subsystem component of the 6800 family designed to provide variable system time intervals.
The EF6840 has three 16-bit binary counters, three corresponding control registers and a status register. These counters are under software control and may be used to cause system interrupts and/or generate output signals. The EF6840 may be utilized for such tasks as frequency measurements, event counting, interval measuring and similar tasks. The device may be used for square wave generation, gated delay signals, single pulses of controlled duration, and pulse width modulation as well as system interrupts.

- Operates from a single 5 V power supply
- Fully TTL compatible
- Single system clock required (Enable)
- Selectable prescaler on timer 3 capable of 4 MHz for the EF6840, 6 MHz for the EF68A40 and 8 MHz for the EF68B40.
- Programmable interrupts ($\overline{IRQ}$) output to MPU
- Readable down counter indicates counts to go to time-out,
- Selectable gating for frequency or pulse-width comparison
- $\overline{RESET}$ input
- Three asynchronous external clock and gate/trigger inputs internally synchronized
- Three maskable outputs
- Three available versions : EF6840 (1.0 MHz)
  EF68A40 (1.5 MHz)
  EF68B40 (2.0 MHz)

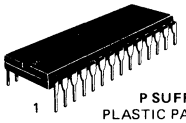
MOS

PROGRAMMABLE TIMER

(N-CHANNEL, SILICON-GATE DEPLETION LOAD)

CASES

CB-132



P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

J SUFFIX
CERDIP PACKAGE

C SUFFIX
CERAMIC PACKAGE

CB-520

FN SUFFIX
PLCC 28

CB-707

E SUFFIX
LCCC 28

Hi-Rel versions available - See chapter 9

FIGURE 1 – PIN ASSIGNMENT

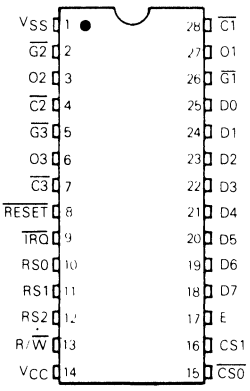
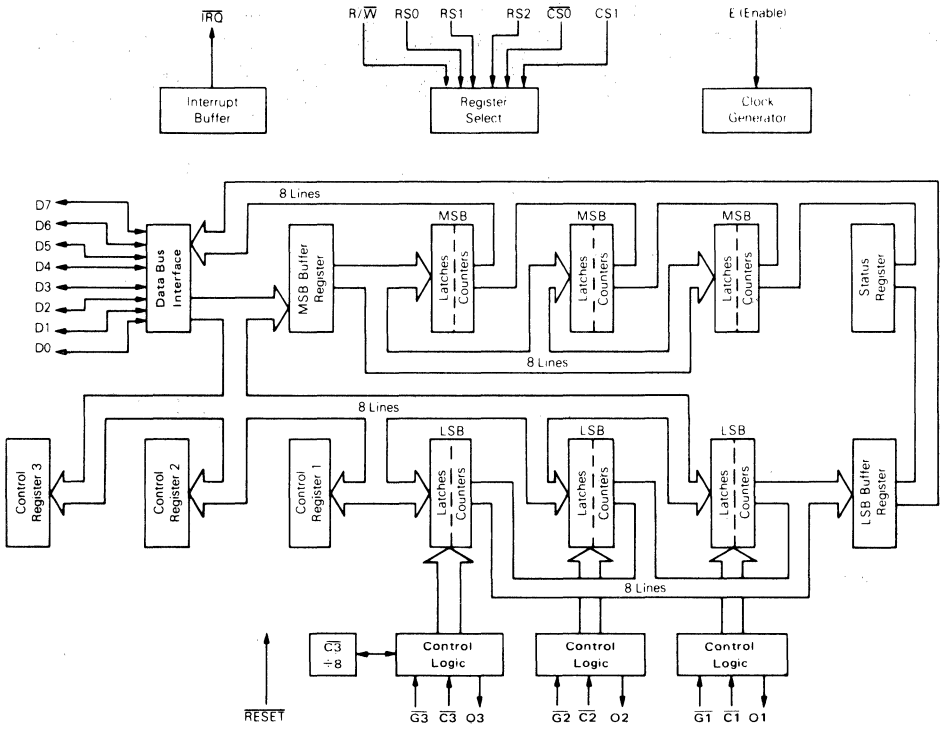


FIGURE 2 – BLOCK DIAGRAM



POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A ≡ Ambient Temperature, °C

θ_{JA} ≡ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts – Chip Internal Power

P_{PORT} ≡ Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K \quad P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range − T_L to T_H EF6840, EF68A40, EF68B40 EF6840, EF68A40, EF68B40 : V suffix EF6840, EF68A40 : M suffix	T_A	0 to +70 −40 to +85 −55 to +125	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Cerdip Plastic Ceramic PLCC	θ_{JA}	 65 115 60 100	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} + 2.0	–	V_{CC}	V
Input Low Voltage		V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)		I_{in}	–	1.0	2.5	μA
Hi-Z (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{Load} = −205 μA) (I_{Load} = −200 μA)	 D0-D7 Other Outputs	V_{OH}	 V_{SS} + 2.4 V_{SS} + 2.4	 – –	 – –	V
Output Low Voltage (I_{Load} = 1.6 mA) (I_{Load} = 3.2 mA)	 $\overline{IRQ}$, D0-D7 O1-O3	V_{OL}	 – –	 – –	 V_{SS} + 0.4 V_{SS} + 0.4	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	470	700	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	 D0-D7 All Others	C_{in}	 – –	 – –	 12.5 7.5	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	 $\overline{IRQ}$ O1, O2, O3	C_{out}	 – –	 – –	 5.0 10	pF

AC OPERATING CHARACTERISTICS (See Figures 4-9)

Characteristic	Symbol	EF6840		EF68A40		EF68B40		Unit
		Min	Max	Min	Max	Min	Max	
Input Rise and Fall Times (Figures 4 and 5) $\overline{C}$, $\overline{G}$, and $\overline{RESET}$	t_r, t_f	–	1.0*	–	0.666*	–	0.500*	μs
Input Pulse Width Low (Figure 4) (Asynchronous Input) $\overline{C}$, $\overline{G}$, and $\overline{RESET}$	PW_L	$t_{cycE} + t_{su} + t_{hd}$	–	$t_{cycE} + t_{su} + t_{hd}$	–	$t_{cycE} + t_{su} + t_{hd}$	–	ns
Input Pulse Width High (Figure 5) (Asynchronous Input) $\overline{C}$, $\overline{G}$	PW_H	$t_{cycE} + t_{su} + t_{hd}$	–	$t_{cycE} + t_{su} + t_{hd}$	–	$t_{cycE} + t_{su} + t_{hd}$	–	ns
Input Setup Time (Figure 6) (Synchronous Input) $\overline{C}$, $\overline{G}$, and $\overline{RESET}$	t_{su}	200	–	120	–	75	–	ns
Input Hold Time (Figure 6) (Synchronous Input) $\overline{C}$, $\overline{G}$, and $\overline{RESET}$	t_{hd}	50		50	–	50	–	ns
Input Synchronization Time (Figure 9) $\overline{C3}$ (−8 Prescaler Mode Only)	t_{sync}	250	–	200	–	175	–	ns
Input Pulse Width $\overline{C3}$ (−8 Prescaler Mode Only)	PW_L, PW_H	120	–	80	–	60	–	ns
Output Delay, O1-O3 (Figure 7) (V_{OH} = 2.4 V, Load B) TTL (V_{OH} = 2.4 V, Load D) MOS (V_{OH} = 0.7 V_{DD}, Load D) CMOS	 t_{CO} t_{CM} t_{CMOS}	 – –	 700 450 2.0	 – – –	 460 450 1.35	 – – –	 340 340 1.0	 ns ns μs
Interrupt Release Time	t_{IR}	–	1.2	–	0.9	–	0.7	μs

*t_r and $t_f \le t_{cycE}$

BUS TIMING CHARACTERISTICS (See Notes 1, 2, and 3)

Ident. Number	Characteristic	Symbol	EF6840		EF68A40		EF68B40		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{CYC}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} max (High Impedance).

FIGURE 3 – BUS TIMING

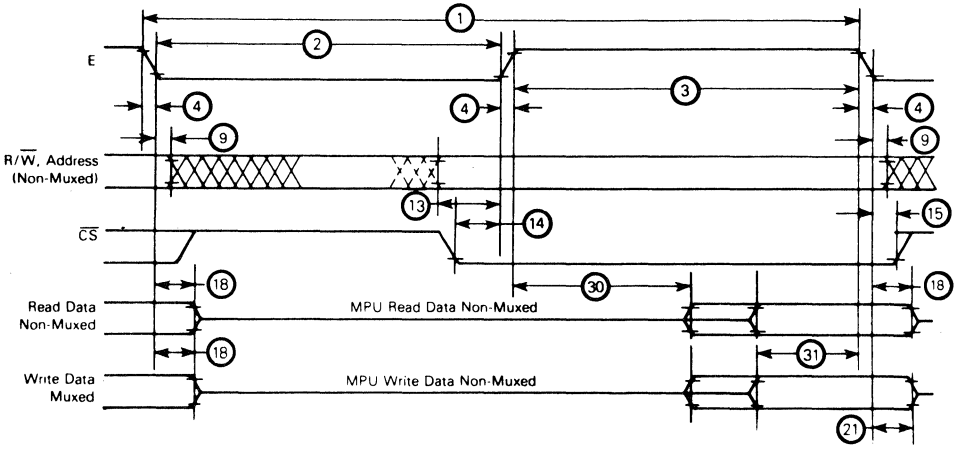


FIGURE 4 – INPUT PULSE WIDTH LOW

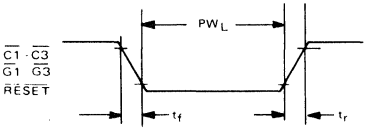
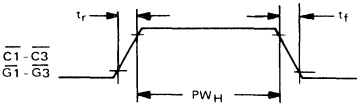


FIGURE 5 – INPUT PULSE WIDTH HIGH



NOTES:
1. Not all signals are applicable to every part.
2. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
3. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

FIGURE 6 – INPUT SETUP AND HOLD TIMES

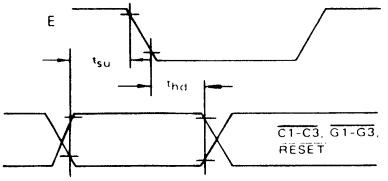


FIGURE 7 – OUTPUT DELAY

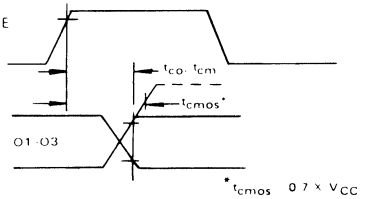


FIGURE 8 – $\overline{IRQ}$ RELEASE TIME

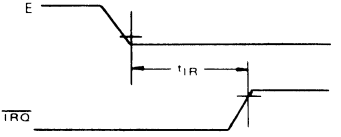


FIGURE 9 – $\overline{C3}$ INPUT SYNCHRONIZATION TIME (–8 PRESCALER MODE ONLY)

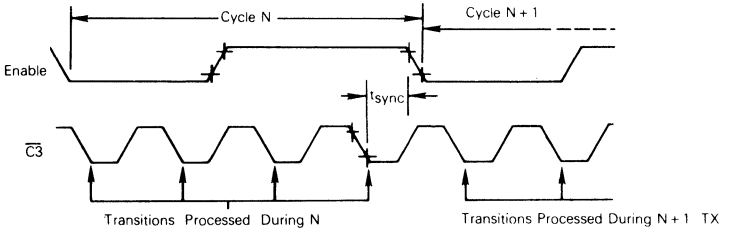
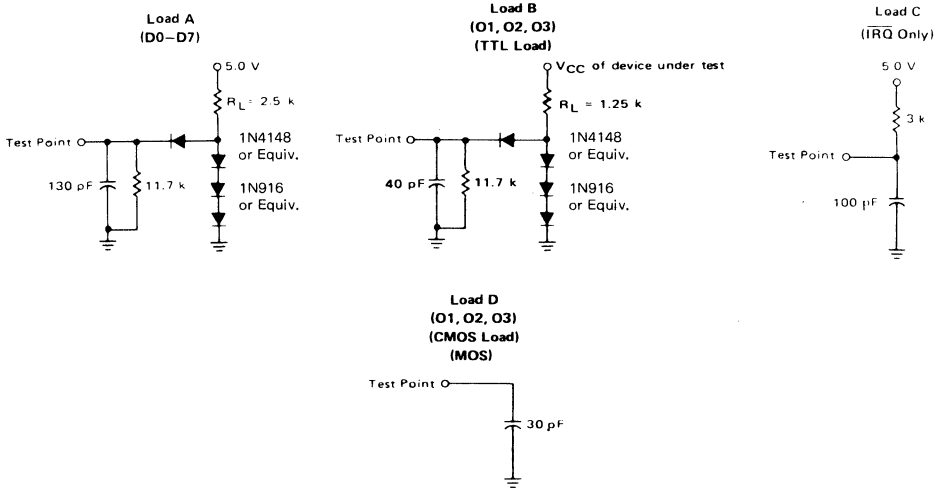


FIGURE 10 – BUS TIMING TEST LOADS



NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.

DEVICE OPERATION

The EF6840 is part of the 6800 microprocessor family and is fully bus compatible with 6800 systems. The three timers in the EF6840 operate independently and in several distinct modes to fit a wide variety of measurement and synthesis applications.

The EF6840 is an integrated set of three distinct counter/timers (Figure 1). It consists of three 16-bit data latches, three 16-bit counters (clocked independently), and the comparison and enable circuitry necessary to implement various measurement and synthesis functions. In addition, it contains interrupt drivers to alert the processor that a particular function has been completed.

In a typical application, a timer will be loaded by first storing two bytes of data into an associated Counter Latch. This data is then transferred into the counter via a Counter Initialization cycle. If the counter is enabled, the counter decrements on each subsequent clock period which may be an external clock, or Enable (E) until one of several predetermined conditions causes it to halt or recycle. The timers are thus programmable, cyclic in nature, controllable by external inputs or the MPU program, and accessible by the MPU at any time.

BUS INTERFACE

The Programmable Timer Module (PTM) interfaces to the 6800 Bus with an 8-bit bidirectional data bus, two Chip Select lines, a Read/Write line, a clock (Enable) line, and Interrupt Request line, an external Reset line, and three Register select lines. VMA should be utilized in conjunction with an MPU address line into a Chip Select of the PTM when using the EF6800/6802/6808.

BIDIRECTIONAL DATA (D0-D7) — The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and PTM. The data bus output drivers are three-state devices which remain in the high-impedance (off) state except when the MPU performs a PTM read operation (Read/Write and Enable lines high and PTM Chip Selects activated).

CHIP SELECT ($\overline{CS0}$, CS1) — These two signals are used to activate the Data Bus interface and allow transfer of data from the PTM. With $\overline{CS0}=0$ and $CS1=1$, the device is selected and data transfer will occur.

READ/WRITE (R/$\overline{W}$) — This signal is generated by the MPU to control the direction of data transfer on the Data Bus. With the PTM selected, a low state on the PTM R/$\overline{W}$ line enables the input buffers and data is transferred from the MPU to the PTM on the trailing edge of the E (Enable) clock. Alternately, (under the same conditions) R/$\overline{W}=1$ and Enable high allows data in the PTM to be read by the MPU.

ENABLE (E CLOCK) — The E clock signal synchronizes data transfer between the MPU and the PTM. It also performs an equivalent synchronization function on the external clock, reset, and gate inputs of the PTM.

INTERRUPT REQUEST ($\overline{IRQ}$) — The active low Interrupt Request signal is normally tied directly (or through priority interrupt circuitry) to the $\overline{IRQ}$ input of the MPU. This is an "open drain" output (no load device on the chip) which permits other similar interrupt request lines to be tied together in a wire-$\overline{OR}$ configuration.

The $\overline{IRQ}$ line is activated if, and only if, the Composite Interrupt Flag (Bit 7 of the Internal Status Register) is asserted. The conditions under which the $\overline{IRQ}$ line is activated are discussed in conjunction with the Status Register.

$\overline{RESET}$ — A low level at this input is clocked into the PTM by the E (Enable) input. Two Enable pulses are required to synchronize and process the signal. The PTM then recognizes the active "low" or inactive "high" on the third Enable pulse. If the $\overline{RESET}$ signal is asynchronous, an additional Enable period is required if setup times are not met. The $\overline{RESET}$ input must be stable High/Low for the minimum time stated in the AC Operating Characteristics.

Recognition of a low level at this input by the PTM causes the following action to occur:

a. All counter latches are preset to their maximum count values.
b. All Control Register bits are cleared with the exception of CR10 (internal reset bit) which is set.
c. All counters are preset to the contents of the latches.
d. All counter outputs are reset and all counter clocks are disabled.
e. All Status Register bits (interrupt flags) are cleared.

REGISTER SELECT LINES (RS0, RS1, $\overline{RS2}$) — These inputs are used in conjunction with the R/$\overline{W}$ line to select the internal registers, counters and latches as shown in Table 1.

NOTE:

The PTM is accessed via MPU Load and Store operations in much the same manner as a memory device. The instructions available with the 6800 family of MPUs which perform read-modify-write operations on memory should not be used when the PTM is accessed. These instructions actually fetch a byte from memory, perform an operation, then restore it to the same address location. Since the PTM uses the R/$\overline{W}$ line as an additional register select input, the modified data will not be restored to the same register if these instructions are used.

CONTROL REGISTER

Each timer in the EF6840 has a corresponding write-only Control Register. Control Register #2 has a unique address space (RS0 = 1, RS = 0, RS2 = 0) and therefore may be written into at any time. The remaining Control Registers (#1 and #3) share the Address Space selected by a logic zero on all Register Select inputs.

CR20 — The least-significant bit of Control Register #2 (CR20) is used as an additional addressing bit for Control Registers #1 and #3. Thus, with all Register selects and R/W inputs at logic zero, Control Register #1 will be written into if CR20 is a logic one. Under the same conditions, Control Register #3 can also be written into after a $\overline{RESET}$ low condition has occurred, since all control register bits (except CR10) are cleared. Therefore, one may write in the sequence CR3, CR2, CR1.

TABLE 1 – REGISTER SELECTION

Register Select Inputs			Operations	
RS2	RS1	RS0	R/$\overline{W}$ = 0	R/$\overline{W}$ = 1
0	0	0	CR20 = 0 Write Control Register #3 CR20 = 1 Write Control Register #1	No Operation
0	0	1	Write Control Register #2	Read Status Register
0	1	0	Write MSB Buffer Register	Read Timer #1 Counter
0	1	1	Write Timer #1 Latches	Read LSB Buffer Register
1	0	0	Write MSB Buffer Register	Read Timer #2 Counter
1	0	1	Write Timer #2 Latches	Read LSB Buffer Register
1	1	0	Write MSB Buffer Register	Read Timer #3 Counter
1	1	1	Write Timer #3 Latches	Read LSB Buffer Register

CR10 – The least significant bit of Control Register #1 is used as an Internal Reset bit. When this bit is a logic zero, all timers are allowed to operate in the modes prescribed by the remaining bits of the control registers. Writing a "one" into CR10 causes all counters to be preset with the contents of the corresponding counter latches, all counter clocks to be disabled, and the timer outputs and interrupt flags (Status Register) to be reset. Counter Latches and Control Registers are undisturbed by an Internal Reset and may be written into regardless of the state of CR10.

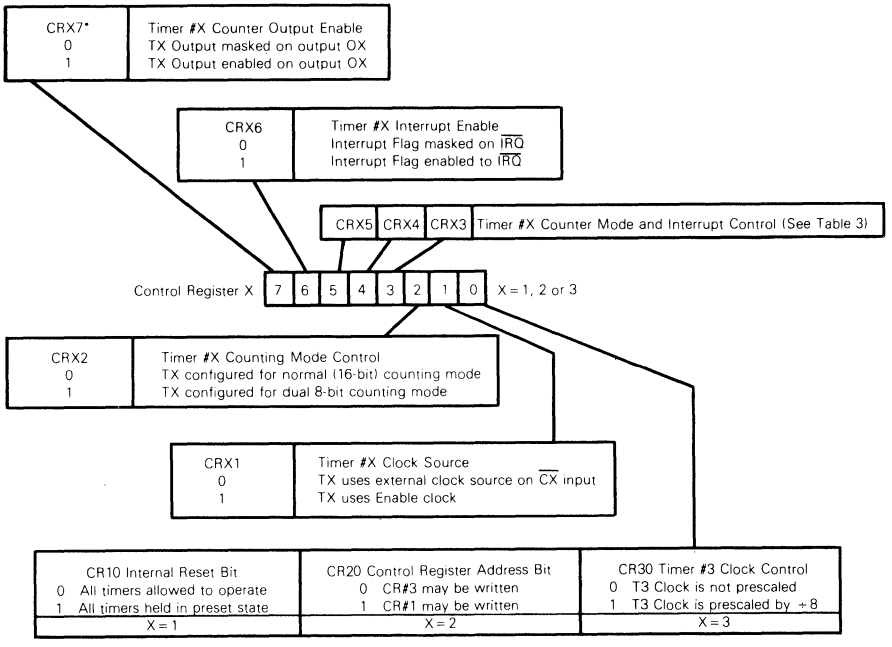
The least signifcant bit of Control Register #3 is used as a selector for a ÷8 prescaler which is available with Timer #3 only. The prescaler, if selected, is effectively placed between the clock input circuitry and the input to Counter #3. It can therefore be used with either the internal clock (Enable) or an external clock source.

NOTE

When initializing Timer 3 into the divide-by-eight mode on consecutive E-cycles (i.e., with DMA), Control Register 3 must be initialized after Timer Latch #3 to insure proper timer initialization.

CR30 – The functions depicted in the foregoing discussions are tabulated in Table 2 for ease of reference.

TABLE 2 – CONTROL REGISTER BITS



Control Register Bits CR10, CR20, and CR30 are unique in that each selects a different function. The remaining bits (1 through 7) of each Control Register select common functions, with a particular Control Register affecting only its corresponding timer.

CRX1 – Bit 1 of Control Register #1 (CR11) selects whether an internal or external clock source is to be used with Timer #1. Similarly, CR21 selects the clock source for Timer #2, and CR31 performs this function for Timer #3. The function of each bit of Control Register "X" can therefore be defined as shown in the remaining section of Table 2.

CRX2 – Control Register Bit 2 selects whether the binary information contained in the Counter Latches (and subsequently loaded into the counter) is to be treated as a single 16-bit word or two 8-bit bytes. In the single 16-bit Counter Mode (CRX2 = 0) the counter will decrement to zero after N + 1 enabled (G = 0) clock periods, where N is defined as the 16-bit number in the Counter Latches. With CRX2 = 1, a similar Time Out will occur after (L + 1)•(M + 1) enabled clock periods, where L and M, respectively, refer to the LSB and MSB bytes in the Counter Latches.

CRX3-CRX7 – Control Register Bits 3, 4, and 5 are explained in detail in the Timer Operating Mode section. Bit 6 is an interrupt mask bit which will be explained more fully in conjunction with the Status Register, and bit 7 is used to enable the corresponding Timer Output. A summary of the control register programming modes is shown in Table 3.

STATUS REGISTER/INTERRUPT FLAGS

The EF6840 has an internal Read-Only Status Register which contains four Interrupt Flags. (The remaining four bits of the register are not used, and defaults to zeros when being read.) Bits 0, 1, and 2 are assigned to Timers 1, 2, and 3, respectively, as individual flag bits, while Bit 7 is a Composite Interrupt Flag. This flag bit will be asserted if any of the individual flag bits is set while Bit 6 of the corresponding Control Register is at a logic one. The conditions for asserting the composite Interrupt Flag bit can therefore be expressed as:

$$INT = I1 \cdot CR16 + I2 \cdot CR26 + I3 \cdot CR36$$

where INT = Composite Interrupt Flag (Bit 7)
I1 = Timer #1 Interrupt Flag (Bit 0)
I2 = Timer #2 Interrupt Flag (Bit 1)
I3 = Timer #3 Interrupt Flag (Bit 2)

An interrupt flag is cleared by a Timer Reset condition, i.e., External $\overline{RESET} = 0$ or Internal Reset Bit (CR10) = 1. It will also be cleared by a Read Timer Counter Command provided that the Status Register has previously been read while the interrupt flag was set. This condition on the Read Status Register-Read Timer Counter (RS-RT) sequence is designed to prevent missing interrupts which might occur after the status register is read, but prior to reading the Timer Counter.

An Individual Interrupt Flag is also cleared by a Write Timer Latches (W) command or a Counter Initialization (CI) sequence, provided that W or CI affects the Timer corresponding to the individual Interrupt Flag.

COUNTER LATCH INITIALIZATION

Each of the three independent timers consists of a 16-bit addressable counter and a 16-bit addressable latch. The counters are preset to the binary numbers stored in the latches. Counter initialization results in the transfer of the latch contents to the counter. See notes in Table 4 regarding the binary number N, L, or M placed into the Latches and their relationship to the output waveforms and counter Time-Outs.

Since the PTM data bus is 8-bits wide and the counters are 16-bits wide, a temporary register (MSB Buffer Register) is provided. This "write only" register is for the Most-Significant Byte of the desired latch data. Three addresses are provided for the MSB Buffer Register (as indicated in Table 1), but they all lead to the same Buffer. Data from the MSB Buffer will automatically be transferred into the Most-Significant Byte of Timer #X when a Write Timer #X Latches Command is performed. So it can be seen that the EF6840 has been designed to allow transfer of two bytes of data into the counter latches provided that the MSB is transferred first. The storage order must be observed to ensure proper latch operation.

In many applications, the source of the data will be a 6800 Family MPU. It should be noted that the 16-bit store operations of the 6800 family microprocessors (STS and STX) transfer data in the order required by the PTM. A Store Index Register Instruction, for example, results in the MSB of the X register being transferred to the selected address, then the LSB of the X register being written into the next higher location. Thus, either the index register or stack pointer may be transferred directly into a selected counter latch with a single instruction.

A logic zero at the $\overline{RESET}$ input also initializes the counter latches. In this case, all latches will assume a maximum count of $65{,}535_{10}$. It is important to note that an Internal

TABLE 3 – PTM OPERATING MODE SELECTION

CRX3	CRX4	CRX5	
0	0	0	Continuous Operating Mode: Gate ↓ or Write to Latches or Reset Causes Counter Initialization
1	0	0	Frequency Comparison Mode: Interrupt If Gate ↑⎍↓ is < Counter Time Out
0	1	0	Continuous Operating Mode: Gate ↓ or Reset Causes Counter Initialization
1	1	0	Pulse Width Comparison Mode: Interrupt if Gate ↑⎍↑ is < Counter Time Out
0	0	1	Single Shot Mode: Gate ↓ or Write to Latches or Reset Causes Counter Initialization
1	0	1	Frequency Comparison Mode: Interrupt If Gate ↑⎍↓ is > Counter Time Out
0	1	1	Single Shot Mode: Gate ↓ or Reset Causes Counter Initialization
1	1	1	Pulse Width Comparison Mode: Interrupt If Gate ↑⎍↑ is > Counter Time Out

Reset (Bit zero of Control Register 1 Set) has no effect on the counter latches.

COUNTER INITIALIZATION

Counter Initialization is defined as the transfer of data from the latches to the counter with subsequent clearing of the Individual Interrupt Flag associated with the counter. Counter Initialization always occurs when a reset condition ($\overline{RESET}$ = 0 or CR10 = 1) is recognized. It can also occur – depending on Timer Mode – with a Write Timer Latches command or recognition of a negative transition of the Gate input.

Counter recycling or re-initialization occurs when a negative transition of the clock input is recognized after the counter has reached an all-zero state. In this case, data is transferred from the Latches to the Counter.

ASYNCHRONOUS INPUT/OUTPUT LINES

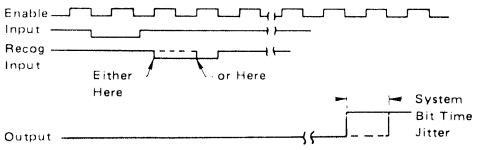
Each of the three timers within the PTM has external clock and gate inputs as well as a counter output line. The inputs are high-impedance, TTL-compatible lines and ouputs are capable of driving two standard TTL loads.

CLOCK INPUTS ($\overline{C1}$, $\overline{C2}$, and $\overline{C3}$) – Input pins $\overline{C1}$, $\overline{C2}$, and $\overline{C3}$ will accept asynchronous TTL voltage level signals to decrement Timers 1, 2, and 3, respectively. The high and low levels of the external clocks must each be stable for at least one system clock period plus the sum of the setup and hold times for the clock inputs. The asynchronous clock rate can vary from dc to the limit imposed by the Enable Clock Setup, and Hold times.

The external clock inputs are clocked in by Enable pulses. Three Enable periods are used to synchronize and process the external clock. The fourth Enable pulse decrements the internal counter. This does not affect the input frequency, it merely creates a delay between a clock input transition and internal recognition of that transition by the PTM. All references to C inputs in this document relate to internal recognition of the input transition. Note that a clock high or low level which does not meet setup and hold time specifications may require an additional Enable pulse for recognition. When observing recurring events, a lack of synchronization will result in "jitter" being observed on the output of the PTM when using asynchronous clocks and gate input signals. There are two types of jitter. "System jitter" is the result of the input signals being out of synchronization with Enable, permitting signals with marginal setup and hold time to be recognized by either the bit time nearest the input transition or the subsequent bit time.

"Input jitter" can be as great as the time between input signal negative going transitions plus the system jitter, if the first transition is recognized during one system cycle, and not recognized the next cycle, or vice versa. See Figure 11.

FIGURE 11 – INPUT JITTER



CLOCK INPUT $\overline{C3}$ (÷8 PRESCALER MODE) – External clock input $\overline{C3}$ represents a special case when Timer #3 is programmed to utilize its optional ÷8 prescaler mode.

The divide-by-8 prescaler contains an asynchronous ripple counter; thus, input setup (t_{su}) and hold times (t_{hd}) do not apply. As long as minimum input pulse widths are maintained, the counter will recognize and process all input clock ($\overline{C3}$) transitions. However, in order to guarantee that a clock transition is processed during the current E cycle, a certain amount of synchronization time (t_{sync}) is required between the $\overline{C3}$ transition and the falling edge of Enable (see Figure 9). If the synchronization time requirement is not met, it is possible that the $\overline{C3}$ transition will not be processed until the following E cycle.

The maximum input frequency and allowable duty cycles for the ÷8 prescaler mode are specified under the AC Operating Characteristics. Internally, the ÷8 prescaler output is treated in the same manner as the previously discussed clock inputs.

GATE INPUTS ($\overline{G1}$, $\overline{G2}$, $\overline{G3}$) – Input pins $\overline{G1}$, $\overline{G2}$, and $\overline{G3}$ accept asynchronous TTL-compatible signals which are used as triggers or clock gating functions to Timers 1, 2, and 3, respectively. The gating inputs are clocked into the PTM by the E (enable) clock in the same manner as the previously discussed clock inputs. That is, a Gate transition is recognized by the PTM on the fourth Enable pulse (provided setup and hold time requirements are met), and the high or low levels of the Gate input must be stable for at least one system clock period plus the sum of setup and hold times. All references to G transition in this document relate to internal recognition of the input transition.

The Gate inputs of all timers directly affect the internal 16-bit counter. The operation of $\overline{G3}$ is therefore independent of the ÷8 prescaler selection.

TIMER OUTPUTS (O1, O2, O3) – Timer outputs O1, O2, and O3 are capable of driving up to two TTL loads and produce a defined output waveform for either Continuous or Single-Shot Timer modes. Output waveform definition is accomplished by selecting either Single 16-bit or Dual 8 bit operating modes. The Single 16-bit mode will produce a square-wave output in the continuous mode and a single pulse in the single-shot mode. The Dual 8-bit mode will produce a variable duty cycle pulse in both the continuous and single-shot timer modes. One bit of each Control Register (CRX7) is used to enable the corresponding output. If this bit is cleared, the output will remain low (V_{OL}) regardless of the operating mode. If it is cleared while the output is high the output will go low during the first enable cycle following a write to the Control Register.

The Continuous and Single-Shot Timer Modes are the only ones for which output response is defined in this data sheet. Refer to the Programmable Timer Fundamentals and Applications manual for a discussion of the output signals in other modes. Signals appear at the outputs (unless CRX7 = 0) during Frequency and Pulse Width comparison modes, but the actual waveform is not predictable in typical applications.

TIMER OPERATING MODES

The EF6840 has been designed to operate effectively in a wide variety of applications. This is accomplished by using three bits of each control register (CRX3, CRX4, and CRX5) to define different operating modes of the Timers. These modes are divided into WAVE SYNTHESIS and WAVE MEASUREMENT modes, and are outlined in Table 4.

TABLE 4 — OPERATING MODES

Control Register			Timer Operating Mode	
CRX3	CRX4	CRX5		
0	*	0	Continuous	Synthesizer
0	*	1	Single-Shot	
1	0	*	Frequency Comparison	Measurement
1	1	*	Pulse Width Comparison	

*Defines Additional Timer Function Selection.

One of the WAVE SYNTHESIS modes is the Continuous Operating mode, which is useful for cyclic wave generation. Either symmetrical or variable duty-cycle waves can be generated in this mode. The other wave synthesis mode, the Single-Shot mode, is similar in use to the Continuous operating mode, however, a single pulse is generated, with a programmable preset width.

The WAVE MEASUREMENT modes include the Frequency Comparison and Pulse Width Comparison modes which are used to measure cyclic and singular pulse widths, respectively.

In addition to the four timer modes in Table 4, the remaining control register bit is used to modify counter initialization and enabling or interrupt conditions.

WAVE SYNTHESIS MODES

CONTINUOUS OPERATING MODE (TABLE 5) — The continuous mode will synthesize a continuous wave with a period proportional to the preset number in the particular timer latches. Any of the timers in the PTM may be programmed to operate in a continuous mode by writing zeroes into bits 3 and 5 of the corresponding control register. Assuming that the timer output is enabled (CRX7 = 1), either a square wave or a variable duty cycle waveform will be generated at the Timer Output, OX. The type of output is selected via Control Register Bit 2.

Either a Timer Reset (CR10 = 1 or External Reset = 0) condition or internal recognition of a negative transition of the Gate input results in Counter Initialization. A Write Timer latches command can be selected as a Counter Initialization signal by clearing CRX4.

The counter is enabled by an absence of a Timer Reset condition and a logic zero at the Gate input. In the 16-bit mode, the counter will decrement on the first clock cycle during or after the counter initialization cycle. It continues to decrement on each clock signal so long as G remains low and no reset condition exists. A Counter Time Out (the first clock after all counter bits = 0) results in the Individual Interrupt Flag being set and reinitialization of the counter.

In the Dual 8-bit mode (CRX2 = 1) [refer to the example in Figure 10 and Tables 5 and 6] the MSB decrements once for every full countdown of the LSB + 1. When the LSB = 0, the MSB is unchanged; on the next clock pulse the LSB is reset to the count in the LSB Latches, and the MSB is decremented by 1 (one). The output, if enabled, remains low during and after initialization and will remain low until the counter MSB is all zeroes. The output will go high at the beginning of the next clock pulse. The output remains high until both the LSB and MSB of the counter are all zeroes. At the beginning of the next clock pulse the defined Time Out (TO) will occur and the output will go low. In the Dual 8-bit mode the period of the output of the example in Figure 12 would span 20 clock pulses as opposed to 1546 clock pulses using the normal 16-bit mode.

A special time-out condition exists for the dual 8-bit mode (CRX2 = 1) if L = 0. In this case, the counter will revert to a mode similar to the single 16-bit mode, except Time Out occurs after M + 1* clock pulses. The output, if enabled, goes low during the Counter Initialization cycle and reverses state at each Time Out. The counter remains cyclical (is reinitialized at each Time Out) and the Individual Interrupt Flag is set when Time Out occurs. If M = L = 0, the internal counters do not change, but the output toggles at a rate of ½ the clock frequency.

TABLE 5 – CONTINUOUS OPERATING MODES

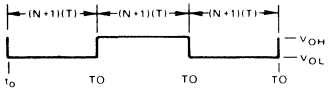
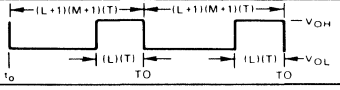
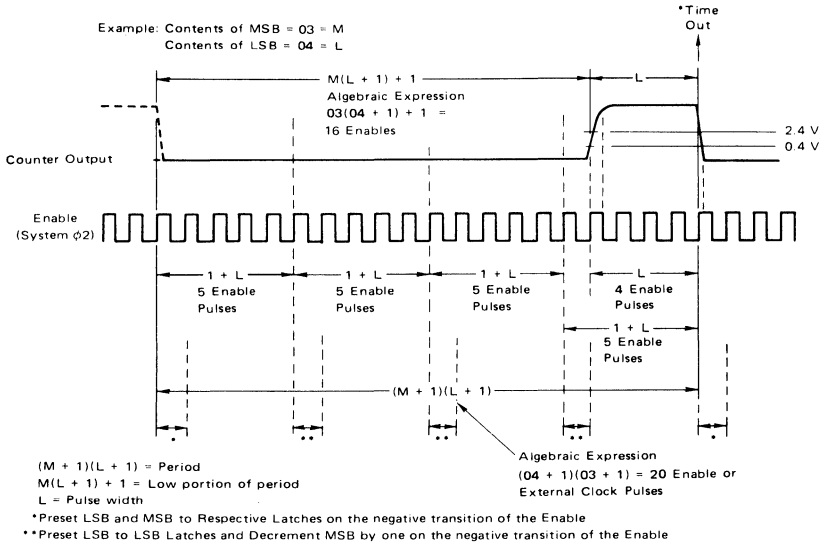
Synthesis Modes		CONTINUOUS MODE (CRX3 = 0, CRX5 = 0)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	*Timer Output (OX) (CRX7 = 1)
0	0	$\overline{G}\downarrow$+W+R	
0	1	$\overline{G}\downarrow$+R	
1	0	$\overline{G}\downarrow$+W+R	
1	1	$\overline{G}\downarrow$+R	

FIGURE 12 – TIMER OUTPUT WAVEFORM EXAMPLE
(Continuous Dual 8-Bit Mode Using Internal Enable)



The discussion of the Continuous Mode has assumed that the application requires an output signal. It should be noted that the Timer operates in the same manner with the output disabled (CRX7=0). A Read Timer Counter command is valid regardless of the state of CRX7.

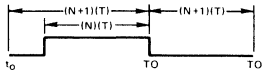
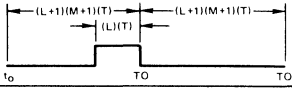
SINGLE-SHOT TIMER MODE – This mode is identical to the Continuous Mode with three exceptions. The first of these is obvious from the name – the output returns to a low level after the initial Time Out and remains low until another Counter Initialization cycle occurs.

As indicated in Table 6, the internal counting mechanism remains cyclical in the Single-Shot Mode. Each Time Out of the counter results in the setting of an Individual Interrupt Flag and re-initialization of the counter.

The second major difference between the Single-Shot and Continuous modes is that the internal counter enable is not dependent on the Gate input level reamining in the low state for the Single-Shot mode.

Another special condition is introduced in the Single-Shot mode. If L=M=0 (Dual 8-bit) or N=0 (Single 16-bit), the output goes low on the first clock received during or after Counter Initialization. The output remains low until the Operating Mode is changed or nonzero data is written into the Counter Latches. Time Outs continue to occur at the end of each clock period.

TABLE 6 – SINGLE-SHOT OPERATING MODES

Synthesis Modes		SINGLE-SHOT MODE (CRX3 = 0, CRX7 = 1, CRX5 = 1)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	Timer Output (OX)
0	0	$\overline{G}\downarrow$+W+R	
0	1	$\overline{G}\downarrow$+R	
1	0	$\overline{G}\downarrow$+W+R	
1	1	$\overline{G}\downarrow$+R	

Symbols are as defined in Table 5.

The three differences between Single-Shot and Continous Timer Mode can be summarized as attributes of the Single-Shot mode:

1. Output is enabled for only one pulse until it is reinitialized.
2. Counter Enable is independent of Gate.
3. L = M = 0 or N = 0 disables output.

Aside from these differences, the two modes are identical.

WAVE MEASUREMENT MODES

TIME INTERVAL MODES – The Time Interval Modes are the Frequency (period) Measurement and Pulse Width Comparison Modes, and are provided for those applications which require more flexibility of interrupt generation and Counter Initialization. Individual Interrupt Flags are set in these modes as a function of both Counter Time Out and transitions of the $\overline{\text{Gate}}$ input. Counter Initialization is also affected by Interrupt Flag status.

A timer's output is normally not used in a Wave Measurement mode, but it is defined. If the output is enabled, it will operate as follows. During the period between reinitialization of the timer and the first Time Out, the output will be a logical zero. If the first Time Out is completed (regardless of its method of generation), the output will go high. If further TO's occur, the output will change state at each completion of a Time-Out.

The counter does operate in either Single 16-bit or Dual 8-bit modes as programmed by CRX2. Other features of the Wave Measurement Modes are outlined in Table 7.

Frequency Comparison Or Period Measurement Mode (CRX3 = 1, CRX4 = 0) – The Frequency Comparison Mode with CRX5 = 1 is straightforward. If Time Out occurs prior to the first negative transition of the $\overline{\text{Gate}}$ input after a Counter Initialization cycle, an Individual Interrupt Flag is set. The counter is disabled, and a Counter Initialization cycle cannot begin until the interrupt flag is cleared and a negative transition on $\overline{\text{G}}$ is detected.

If CRX5 = 0, as shown in Tables 7 and 8, an interrupt is generated if $\overline{\text{Gate}}$ input returns low prior to a Time Out. If a Counter Time Out occurs first, the counter is recycled and continues to decrement. A bit is set within the timer on the initial Time Out which precludes further individual interrupt generation until a new Counter Initialization cycle has been completed. When this internal bit is set, a negative transition of the $\overline{\text{Gate}}$ input starts a new Counter Initialization cycle. (The condition of $\overline{\text{G}} \downarrow \cdot \overline{\text{T}} \cdot \text{TO}$ is satisfied, since a Time Out has occurred and no individual Interrupt has been generated.)

Any of the timers within the PTM may be programmed to compare the period of a pulse (giving the frequency after calculations) at the $\overline{\text{Gate}}$ input with the time period requested for Counter Time Out. A negative transition of the $\overline{\text{Gate}}$ Input enables the counter and starts a Counter Initialization cycle – provided that other conditions, as noted in Table 8, are satisfied. The counter decrements on each clock signal recognized during or after Counter Initialization until an Interrupt is generated, a Write Timer Latches command is issued, or a Timer Reset condition occurs. It can be seen from Table 8 that an interrupt condition will be generated if CRX5 = 0 and the period of the pulse (single pulse or measured separately repetitive pulses) at the Gate input is less than the Counter Time Out period. If CRX5 = 1, an interrupt is generated if the reverse is true.

Assume now with CRX5 = 1 that a Counter Initialization has occurred and that the Gate input has returned low prior to Counter Time Out. Since there is no Individual Interrupt Flag generated, this automatically starts a new Counter Initialization Cycle. The process will continue with frequency comparison being performed on each Gate input cycle until the mode is changed, or a cycle is determined to be above the predetermined limit.

Pulse Width Comparison Mode (CRX3 = 1, CRX4 = 1) – This mode is similar to the Frequency Comparison Mode except for a positive, rather than negative, transition of the Gate input terminates the count. With CRX5 = 0, an Individual Interrupt Flag will be generated if the zero level pulse applied to the Gate input is less than the time period required for Counter Time Out. With CRX5 = 1, the interrupt is generated when the reverse condition is true.

As can be seen in Table 8, a positive transition of the Gate input disables the counter. With CRX5 = 0, it is therefore possible to directly obtain the width of any pulse causing an interrupt. Similar data for other Time Interval Modes and conditions can be obtained, if two sections of the PTM are dedicated to the purpose.

FIGURE 7 – OUTPUT DELAY

CRX3 = 1			
CRX4	**CRX5**	**Application**	**Condition for Setting Individual Interrupt Flag**
0	0	Frequency Comparison	Interrupt Generated if $\overline{\text{Gate}}$ Input Period (1/F) is less than Counter Time Out (TO)
0	1	Frequency Comparison	Interrupt Generated if $\overline{\text{Gate}}$ Input Period (1/F) is greater than Counter Time Out (TO)
1	0	Pulse Width Comparison	Interrupt Generated if $\overline{\text{Gate}}$ Input "Down Time" is less than Counter Time Out (TO)
1	1	Pulse Width Comparison	Interrupt Generated if $\overline{\text{Gate}}$ Input "Down Time" is greater than Counter Time Out (TO)

TABLE 8 – FREQUENCY COMPARISON MODE

Mode	Bit 3	Bit 4	Control Reg. Bit 5	Counter Initialization	Counter Enable Flip-Flop Set (CE)	Counter Enable Flip-Flop Reset (CE)	Interrupt Flag Set (I)
Frequency	1	0	0	$\overline{G\uparrow} \cdot I + \overline{(CE + TO)} + R$	$\overline{G\uparrow} \cdot \overline{W} \cdot \overline{R} \cdot \overline{T}$	$W \cdot R \cdot I$	$G\uparrow$ Before TO
Comparison	1	0	1	$\overline{G\uparrow} \cdot \overline{T} + R$	$\overline{G\uparrow} \cdot \overline{W} \cdot \overline{R} \cdot \overline{T}$	$W \cdot R \cdot I$	TO Before $\overline{G\uparrow}$
Pulse Width	1	1	0	$\overline{G\uparrow} \cdot \overline{T} + \overline{R}$	$\overline{G\uparrow} \overline{W} \cdot \overline{R} \cdot \overline{T}$	$W \cdot R \cdot I \cdot G$	$\overline{G\uparrow}$ Before TO
Comparison	1	1	1	$\overline{G\uparrow} \cdot \overline{T} + \overline{R}$	$\overline{G\uparrow} \cdot \overline{W} \cdot \overline{R} \cdot \overline{T}$	$W \cdot R \cdot I \cdot G$	$\overline{G\uparrow}$ Before TO

$\overline{G\uparrow}$ = Negative transition of $\overline{\text{Gate}}$ input.
W = Write Timer Latches Command
R = Timer Reset (CR10 = 1 or External $\overline{RESET}$ = 0)
N = 16-Bit Number in Counter Latch.
TO = Counter Time Out (All Zero Condition)
I = Interrupt for a given timer

*All time intervals shown above assume the Gate ($\overline{G}$) and Clock ($\overline{C}$) signals are synchronized to the system clock (E) with the specified setup and hold time requirements

ORDERING INFORMATION

EF68A40 | C | M | B/B

- EF68A40: Device
- C: Package
- M: Oper. temp.
- B/B: Screening level

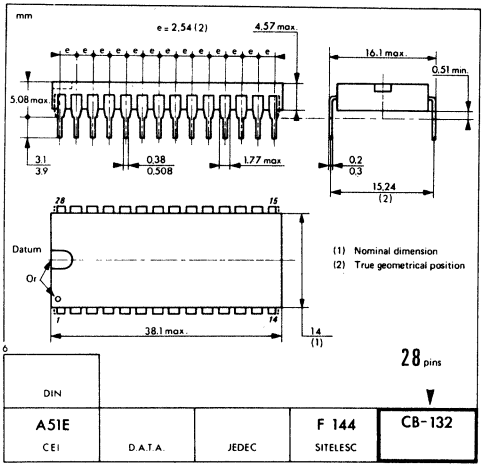
The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

DEVICE	PACKAGE					OPER. TEMP			SCREENING LEVEL			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6840 (1.0 MHz)	●	●	●		●	●			●			
	●	●	●				●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68A40 (1.5 MHz)	●	●	●			●			●			
	●	●	●				●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68B40 (2.0 MHz)	●	●	●			●			●			
	●	●					●		●		●	

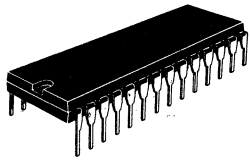
Examples : EF6840C, EF6840CV, EF6840CM, EF6840JM

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** −40°C to +85°C, **M :** −55°C to +125°C, ***** : may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.

PHYSICAL DIMENSIONS



CB-132



P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

J SUFFIX
CERDIP PACKAGE

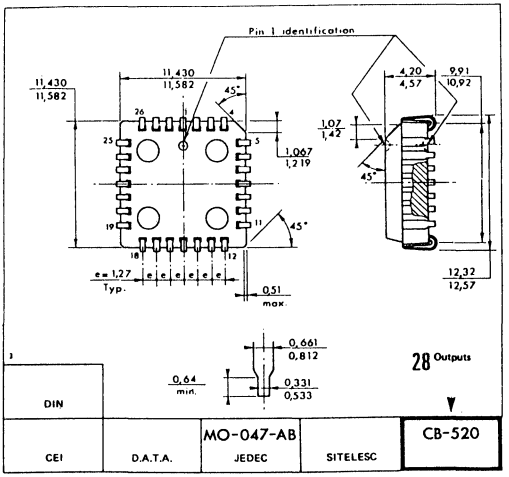
C SUFFIX
CERAMIC PACKAGE

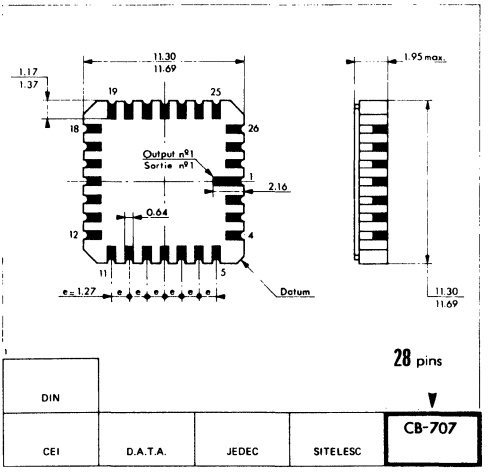
CB-520

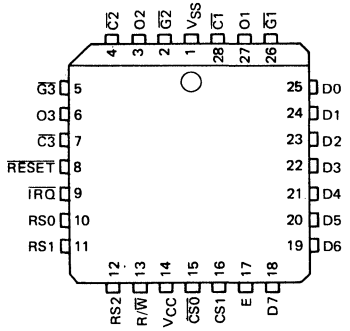
FN SUFFIX
PLCC 28

CB-707

E SUFFIX
LCCC 28







THOMSON COMPONENTS MOSTEK

EF6850

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The EF6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the EF6800 Microprocessing Unit.

The bus interface of the EF6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation, three control lines are provided.

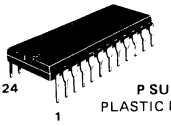
- 8- and 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional ÷1, ÷16, and ÷64 Clock Modes
- Up to 1.0 Mbps Transmission
- False Start Bit Deletion
- Peripheral/Modem Control Functions
- Double Buffered
- One- or Two-Stop Bit Operation

MOS

(N-CHANNEL, SILICON-GATE)

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER

CB-68 **CASES**



P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

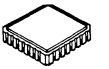
J SUFFIX
CERDIP PACKAGE

CB-520

FN SUFFIX
PLCC 28

C SUFFIX
CERAMIC PACKAGE

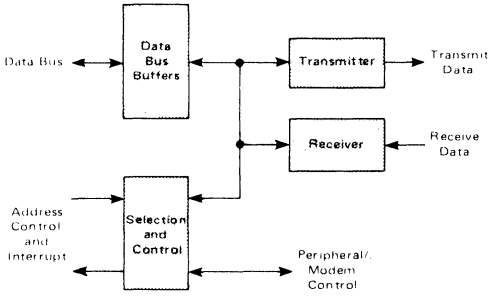
CB-707



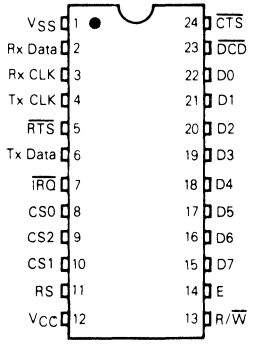
E SUFFIX
LCCC 28

Hi-Rel versions available - See chapter 9

EF6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT



MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range EF6850, EF68A50, EF68B50 EF6850, EF68A50, EF68B50 : V suffix EF6850, EF68A50 : M suffix	T_A	T_L to T_H 0 to 70 −40 to +85 −55 to +125	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Ceramic Cerdip **PLCC**	θ_{JA}	 120 60 65 100	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts – Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/$\overline{W}$, CS0, CS1, $\overline{CS2}$, Enable RS, Rx D, Rx C, $\overline{CTS}$, $\overline{DCD}$	I_{in}	–	1.0	2.5	μA
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{Load} = −205 μA, Enable Pulse Width < 25 μs) (I_{Load} = −100 μA, Enable Pulse Width < 25 μs)	D0-D7 Tx Data, $\overline{RTS}$	V_{OH}	$V_{SS}+2.4$ $V_{SS}+2.4$	– –	– –	V
Output Low Voltage (I_{Load} = 1.6 mA, Enable Pulse Width < 25 μs)		V_{OL}	–	–	$V_{SS}+0.4$	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	–	300	525*	mW
Internal Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz) E, Tx CLK, Rx CLK R/$\overline{W}$, RS, Rx Data, CS0, CS1, $\overline{CS2}$, $\overline{CTS}$, $\overline{DCD}$	D0-D7	C_{in}	– –	10 7.0	12.5 7.5	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	RTS, Tx Data $\overline{IRQ}$	C_{out}	– –	– –	10 5.0	pF

* For temperatures less than T_A = 0°C, P_{INT} maximum will increase.

SERIAL DATA TIMING CHARACTERISTICS

Characteristic		Symbol	EF6850		EF68A50		EF68B50		Unit
			Min	Max	Min	Max	Min	Max	
Data Clock Pulse Width, Low (See Figure 1)	÷16, ÷64 Modes	PW_{CL}	600	–	450	–	280	–	ns
	÷1 Mode		900	–	650	–	500	–	
Data Clock Pulse Width, High (See Figure 2)	÷16, ÷64 Modes	PW_{CH}	600	–	450	–	280	–	ns
	÷1 Mode		900	–	650	–	500	–	
Data Clock Frequency	÷16, ÷64 Modes	f_C	–	0.8	–	1.0	–	1.5	MHz
	÷1 Mode		–	500	–	750	–	1000	kHz
Data Clock-to-Data Delay for Transmitter (See Figure 3)		t_{TDD}	–	600	–	540	–	460	ns
Receive Data Setup Time (See Figure 4)	÷1 Mode	t_{RDS}	250	–	100	–	30	–	ns
Receive Data Hold Time (See Figure 5)	÷1 Mode	t_{RDH}	250	–	100	–	30	–	ns
Interrupt Request Release Time (See Figure 6)		t_{IR}	–	1.2	–	0.9	–	0.7	μs
Request-to-Send Delay Time (See Figure 6)		t_{RTS}	–	560	–	480	–	400	ns
Input Rise and Fall Times (or 10% of the pulse width if smaller)		t_r, t_f	–	1.0	–	0.5	–	0.25	μs

FIGURE 1 – CLOCK PULSE WIDTH, LOW-STATE

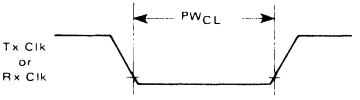


FIGURE 2 – CLOCK PULSE WIDTH, HIGH-STATE

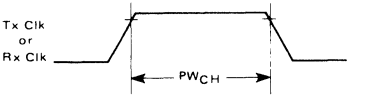


FIGURE 3 – TRANSMIT DATA OUTPUT DELAY

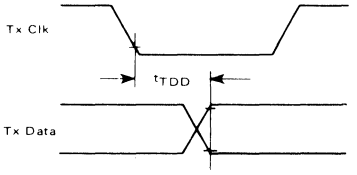


FIGURE 4 – RECEIVE DATA SETUP TIME (÷1 Mode)

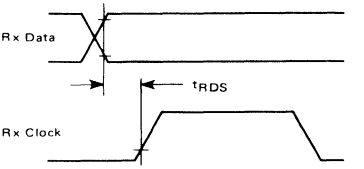


FIGURE 5 – RECEIVE DATA HOLD TIME (÷1 Mode)

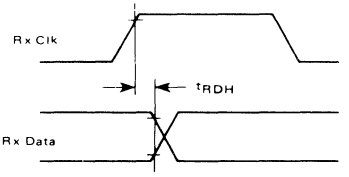
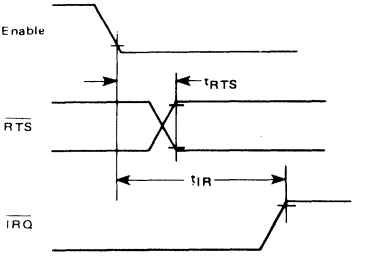


FIGURE 6 – REQUEST-TO-SEND DELAY AND INTERRUPT-REQUEST RELEASE TIMES



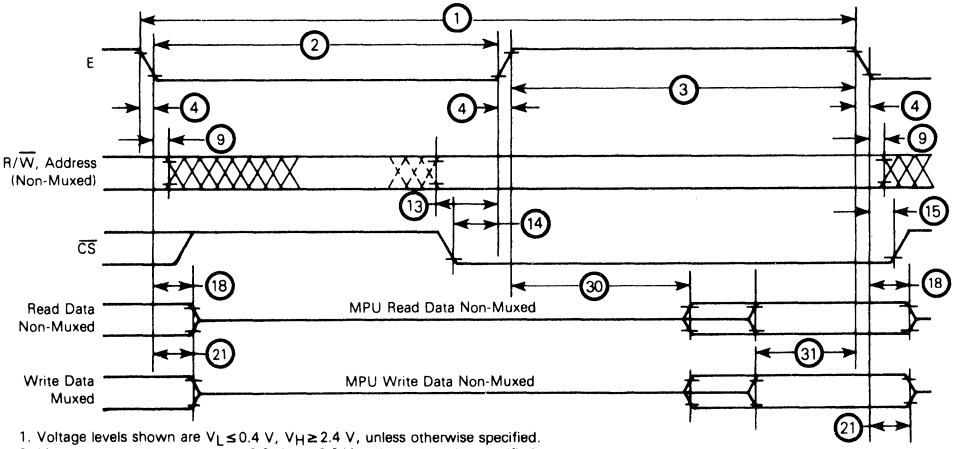
Note: Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.

BUS TIMING CHARACTERISTICS (See Notes 1 and 2 and Figure 7)

Ident. Number	Characteristic	Symbol	EF6850		EF68A50		EF68B50		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

FIGURE 7 — BUS TIMING CHARACTERISTICS



1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

FIGURE 8 — BUS TIMING TEST LOADS

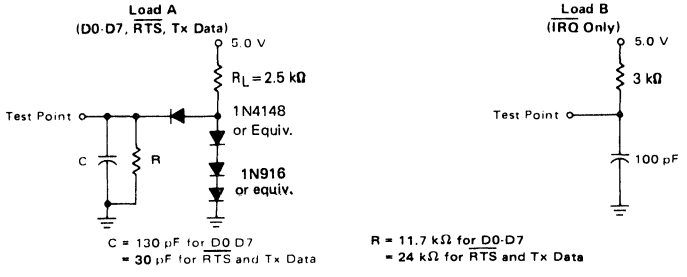
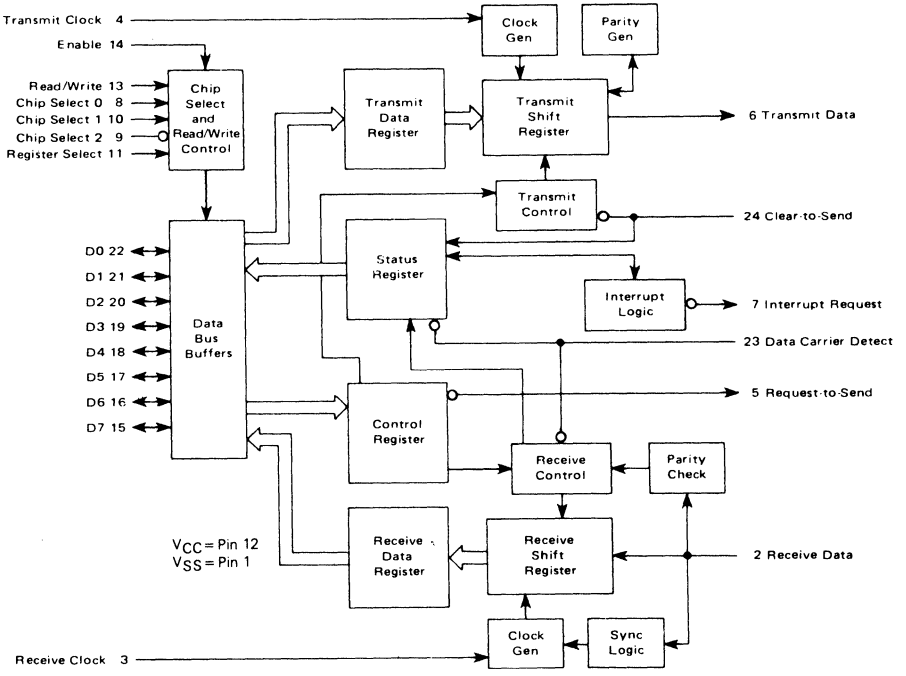


FIGURE 9 – EXPANDED BLOCK DIAGRAM



DEVICE OPERATION

At the bus interface, the ACIA appears as two addressable memory locations. Internally, there are four registers: two read-only and two write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and three peripheral/modem control lines.

POWER ON/MASTER RESET

The master reset (CR0, CR1) should be set during system initialization to insure the reset condition and prepare for programming the ACIA functional configuration when the communications channel is required. During the first master reset, the $\overline{IRQ}$ and $\overline{RTS}$ outputs are held at level 1. On all other master resets, the $\overline{RTS}$ output can be programmed high or low with the $\overline{IRQ}$ output held high. Control bits CR5 and CR6 should also be programmed to define the state of $\overline{RTS}$ whenever master reset is utilized. The ACIA also contains internal power-on reset logic to detect the power line turn-on transition and hold the chip in a reset state to prevent erroneous output transitions prior to initialization. This circuitry depends on clean power turn-on transitions. The power-on reset is released by means of the bus-programmed master reset which must be applied prior to operating the ACIA. After master resetting the ACIA, the programmable Control Register can be set for a number of options such as variable clock divider ratios, variable word-length, one or two stop bits, parity (even, odd, or none), etc.

TRANSMIT

A typical transmitting sequence consists of reading the ACIA Status Register either as a result of an interrupt or in the ACIA's turn in a polling sequence. A character may be written into the Transmit Data Register if the status read operation has indicated that the Transmit Data Register is empty. This character is transferred to a Shift Register where it is serialized and transmitted from the Transmit Data output preceded by a start bit and followed by one or two stop bits. Internal parity (odd or even) can be optionally added to the character and will occur between the last data bit and the first stop bit. After the first character is written in the Data Register, the Status Register can be read again to check for a Transmit Data Register Empty condition and current peripheral status. If the register is empty, another character can be loaded for transmission even though the first character is in the process of being transmitted (because of

double buffering). The second character will be automatically transferred into the Shift Register when the first character transmission is completed. This sequence continues until all the characters have been transmitted.

RECEIVE

Data is received from a peripheral by means of the Receive Data input. A divide-by-one clock ratio is provided for an externally synchronized clock (to its data) while the divide-by-16 and 64 ratios are provided for internal synchronization. Bit synchronization in the divide-by-16 and 64 modes is initiated by the detection of 8 or 32 low samples on the receive line in the divide-by-16 and 64 modes respectively. False start bit deletion capability insures that a full half bit of a start bit has been received before the internal clock is synchronized to the bit time. As a character is being received, parity (odd or even) will be checked and the error indication will be available in the Status Register along with framing error, overrun error, and Receive Data Register full. In a typical receiving sequence, the Status Register is read to determine if a character has been received from a peripheral. If the Receiver Data Register is full, the character is placed on the 8-bit ACIA bus when a Read Data command is received from the MPU. When parity has been selected for a 7-bit word (7 bits plus parity), the receiver strips the parity bit (D7 = 0) so that data alone is transferred to the MPU. This feature reduces MPU programming. The Status Register can continue to be read to determine when another character is available in the Receive Data Register. The receiver is also double buffered so that a character can be read from the data register as another character is being received in the shift register. The above sequence continues until all characters have been received.

INPUT/OUTPUT FUNCTIONS

ACIA INTERFACE SIGNALS FOR MPU

The ACIA interfaces to the 6800 MPU with an 8-bit bidirectional data bus, three chip select lines, a register select line, an interrupt request line, read/write line, and enable line. These signals permit the MPU to have complete control over the ACIA.

ACIA Bidirectional Data (D0-D7) — The bidirectional data lines (D0-D7) allow for data transfer between the ACIA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ACIA read operation.

ACIA Enable (E) — The Enable signal, E, is a high-impedance TTL-compatible input that enables the bus input/output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the E F6800 $\phi 2$ Clock or EF6809E clock.

Read/Write (R/$\overline{W}$) — The Read/Write line is a high-impedance input that is TTL compatible and is used to control the direction of data flow through the ACIA's input/output data bus interface. When Read/Write is high (MPU Read cycle), ACIA output drivers are turned on and a selected register is read. When it is low, the ACIA output drivers are turned off and the MPU writes into a selected register. Therefore, the Read/Write signal is used to select read-only or write-only registers within the ACIA.

Chip Select (CS0, CS1, $\overline{CS2}$) — These three high-impedance TTL-compatible input lines are used to address the ACIA. The ACIA is selected when CS0 and CS1 are high and $\overline{CS2}$ is low. Transfers of data to and from the ACIA are then performed under the control of the Enable Signal, Read/Write, and Register Select.

Register Select (RS) — The Register Select line is a high-impedance input that is TTL compatible. A high level is used to select the Transmit/Receive Data Registers and a low level the Control/Status Registers. The Read/Write signal line is used in conjunction with Register Select to select the read-only or write-only register in each register pair.

Interrupt Request ($\overline{IRQ}$) — Interrupt Request is a TTL-compatible, open-drain (no internal pullup), active low output that is used to interrupt the MPU. The $\overline{IRQ}$ output remains low as long as the cause of the interrupt is present and the appropriate interrupt enable within the ACIA is set. The $\overline{IRQ}$ status bit, when high, indicates the $\overline{IRQ}$ output is in the active state.

Interrupts result from conditions in both the transmitter and receiver sections of the ACIA. The transmitter section causes an interrupt when the Transmitter Interrupt Enabled condition is selected (CR5•$\overline{CR6}$), and the Transmit Data Register Empty (TDRE) status bit is high. The TDRE status bit indicates the current status of the Transmitter Data Register except when inhibited by Clear-to-Send ($\overline{CTS}$) being high or the ACIA being maintained in the Reset condition. The interrupt is cleared by writing data into the Transmit Data Register. The interrupt is masked by disabling the Transmitter Interrupt via CR5 or CR6 or by the loss of $\overline{CTS}$ which inhibits the TDRE status bit. The Receiver section causes an interrupt when the Receiver Interrupt Enable is set and the Receive Data Register Full (RDRF) status bit is high, an Overrun has occurred, or Data Carrier Detect ($\overline{DCD}$) has gone high. An interrupt resulting from the RDRF status bit can be cleared by reading data or resetting the ACIA. Interrupts caused by Overrun or loss of $\overline{DCD}$ are cleared by reading the status register after the error condition has occurred and then reading the Receive Data Register or resetting the ACIA. The receiver interrupt is masked by resetting the Receiver Interrupt Enable.

CLOCK INPUTS

Separate high-impedance TTL-compatible inputs are provided for clocking of transmitted and received data. Clock frequencies of 1, 16, or 64 times the data rate may be selected.

Transmit Clock (Tx CLK) — The Transmit Clock input is used for the clocking of transmitted data. The transmitter initiates data on the negative transition of the clock.

Receive Clock (Rx CLK) — The Receive Clock input is used for synchronization of received data. (In the ÷ 1 mode, the clock and data must be synchronized externally.) The receiver samples the data on the positive transition of the clock.

SERIAL INPUT/OUTPUT LINES

Receive Data (Rx Data) – The Receive Data line is a high-impedance TTL-compatible input through which data is received in a serial format. Synchronization with a clock for detection of data is accomplished internally when clock rates of 16 or 64 times the bit rate are used.

Transmit Data (Tx Data) – The Transmit Data output line transfers serial data to a modem or other peripheral.

PERIPHERAL/MODEM CONTROL

The ACIA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Request-to-Send and Data Carrier Detect.

Clear-to-Send ($\overline{CTS}$) – This high-impedance TTL-compatible input provides automatic control of the transmitting end of a communications link via the modem Clear-to-Send active low output by inhibiting the Transmit Data Register Empty (TDRE) status bit.

Request-to-Send ($\overline{RTS}$) – The Request-to-Send output enables the MPU to control a peripheral or modem via the data bus. The RTS output corresponds to the state of the Control Register bits CR5 and CR6. When CR6 = 0 or both CR5 and CR6 = 1, the $\overline{RTS}$ output is low (the active state). This output can also be used for Data Terminal Ready (DTR).

Data Carrier Detect ($\overline{DCD}$) – This high-impedance TTL-compatible input provides automatic control, such as in the receiving end of a communications link by means of a modem Data Carrier Detect output. The $\overline{DCD}$ input inhibits and initializes the receiver section of the ACIA when high. A low-to-high transition of the Data Carrier Detect initiates an interrupt to the MPU to indicate the occurrence of a loss of carrier when the Receive Interrupt Enable bit is set. The Rx CLK must be running for proper $\overline{DCD}$ operation.

ACIA REGISTERS

The expanded block diagram for the ACIA indicates the internal registers on the chip that are used for the status, control, receiving, and transmitting of data. The content of each of the registers is summarized in Table 1.

TRANSMIT DATA REGISTER (TDR)

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the ACIA has been addressed with RS high and R/$\overline{W}$ low. Writing data into the register causes the Transmit Data Register Empty bit in the Status Register to go low. Data can then be transmitted. If the transmitter is idling and no character is being transmitted, then the transfer will take place within 1-bit time of the trailing edge of the Write command. If a character is being transmitted, the new data character will commence as soon as the previous character is complete. The transfer of data causes the Transmit Data Register Empty (TDRE) bit to indicate empty.

RECEIVE DATA REGISTER (RDR)

Data is automatically transferred to the empty Receive Data Register (RDR) from the receiver deserializer (a shift register) upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRF) in the status buffer to go high (full). Data may then be read through the bus by addressing the ACIA and selecting the Receive Data Register with RS and R/$\overline{W}$ high when the ACIA is enabled. The non-destructive read cycle causes the RDRF bit to be cleared to empty although the data is retained in the RDR. The status is maintained by RDRF as to whether or not the data is current. When the Receive Data Register is full, the automatic transfer of data from the Receiver Shift Register to the Data Register is inhibited and the RDR contents remain valid with its current status stored in the Status Register.

TABLE 1 – DEFINITION OF ACIA REGISTER CONTENTS

Data Bus Line Number	Buffer Address			
	RS • R/$\overline{W}$ Transmit Data Register (Write Only)	RS • R/$\overline{W}$ Receive Data Register (Read Only)	$\overline{RS}$ • R/$\overline{W}$ Control Register (Write Only)	$\overline{RS}$ • R/$\overline{W}$ Status Register (Read Only)
0	Data Bit 0*	Data Bit 0	Counter Divide Select 1 (CR0)	Receive Data Register Full (RDRF)
1	Data Bit 1	Data Bit 1	Counter Divide Select 2 (CR1)	Transmit Data Register Empty (TDRE)
2	Data Bit 2	Data Bit 2	Word Select 1 (CR2)	Data Carrier Detect ($\overline{DCD}$)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear-to-Send ($\overline{CTS}$)
4	Data Bit 4	Data Bit 4	Word Select 3 (CR4)	Framing Error (FE)
5	Data Bit 5	Data Bit 5	Transmit Control 1 (CR5)	Receiver Overrun (OVRN)
6	Data Bit 6	Data Bit 6	Transmit Control 2 (CR6)	Parity Error (PE)
7	Data Bit 7***	Data Bit 7**	Receive Interrupt Enable (CR7)	Interrupt Request ($\overline{IRQ}$)

* Leading bit = LSB = Bit 0

** Data bit will be zero in 7 bit plus parity modes

*** Data bit is "don't care" in 7 bit plus parity modes.

CONTROL REGISTER

The ACIA Control Register consists of eight bits of write-only buffer that are selected when RS and R/$\overline{W}$ are low. This register controls the function of the receiver, transmitter, interrupt enables, and the Request-to-Send peripheral/modem control output.

Counter Divide Select Bits (CR0 and CR1) – The Counter Divide Select Bits (CR0 and CR1) determine the divide ratios utilized in both the transmitter and receiver sections of the ACIA. Additionally, these bits are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on $\overline{CTS}$ and $\overline{DCD}$) and initializes both the receiver and transmitter. Master reset does not affect other Control Register bits. Note that after power-on or a power fail/restart, these bits must be set high to reset the ACIA. After resetting, the clock divide ratio may be selected. These counter select bits provide for the following clock divide ratios:

CR1	CR0	Function
0	0	÷1
0	1	÷16
1	0	÷64
1	1	Master Reset

Word Select Bits (CR2, CR3, and CR4) – The Word Select bits are used to select word length, parity, and the number of stop bits. The encoding format is as follows:

CR4	CR3	CR2	Function
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bit
0	1	1	7 Bits + Odd Parity + 1 Stop Bit
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bit
1	1	0	8 Bits + Even parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit

Word length, Parity Select, and Stop Bit changes are not buffered and therefore become effective immediately.

Transmitter Control Bits (CR5 and CR6) – Two Transmitter Control bits provide for the control of the interrupt from the Transmit Data Register Empty condition, the Request-to-Send ($\overline{RTS}$) output, and the transmission of a Break level (space). The following encoding format is used:

CR6	CR5	Function
0	0	$\overline{RTS}$ = low, Transmitting Interrupt Disabled.
0	1	$\overline{RTS}$ = low, Transmitting Interrupt Enabled.
1	0	$\overline{RTS}$ = high, Transmitting Interrupt Disabled.
1	1	$\overline{RTS}$ = low, Transmits a Break level on the Transmit Data Output. Transmitting Interrupt Disabled.

Receive Interrupt Enable Bit (CR7) – The following interrupts will be enabled by a high level in bit position 7 of the Control Register (CR7): Receive Data Register Full, Overrun, or a low-to-high transition on the Data Carrier Detect ($\overline{DCD}$) signal line.

STATUS REGISTER

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. This read-only register is selected when RS is low and R/$\overline{W}$ is high. Information stored in this register indicates the status of the Transmit Data Register, the Receive Data Register and error logic, and the peripheral/modem status inputs of the ACIA.

Receive Data Register Full (RDRF), Bit 0 – Receive Data Register Full indicates that received data has been transferred to the Receive Data Register. RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current. Data Carrier Detect being high also causes RDRF to indicate empty.

Transmit Data Register Empty (TDRE), Bit 1 – The Transmit Data Register Empty bit being set high indicates that the Transmit Data Register contents have been transferred and that new data may be entered. The low state indicates that the register is full and that transmission of a new character has not begun since the last write data command.

Data Carrier Detect ($\overline{DCD}$), Bit 2 – The Data Carrier Detect bit will be high when the $\overline{DCD}$ input from a modem has gone high to indicate that a carrier is not present. This bit going high causes an Interrupt Request to be generated when the Receive Interrupt Enable is set. It remains high after the $\overline{DCD}$ input is returned low until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the $\overline{DCD}$ input remains high after read status and read data or master reset has occurred, the interrupt is cleared, the $\overline{DCD}$ status bit remains high and will follow the $\overline{DCD}$ input.

Clear-to-Send ($\overline{CTS}$), Bit 3 – The Clear-to-Send bit indicates the state of the Clear-to-Send input from a modem. A low $\overline{CTS}$ indicates that there is a Clear-to-Send from the modem. In the high state, the Transmit Data Register Empty bit is inhibited and the Clear-to-Send status bit will be high. Master reset does not affect the Clear-to-Send status bit.

Framing Error (FE), Bit 4 – Framing error indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the first stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

Receiver Overrun (OVRN), Bit 5 – Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun condition begins at the midpoint of the last bit of the second character received in succession without a read of the RDR having occurred. The Overrun does not occur in the Status Register until the valid character prior to Overrun has

been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

Parity Error (PE), Bit 6 – The parity error flag indicates that the number of highs (ones) in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver parity check results are inhibited.

Interrupt Request ($\overline{IRQ}$), Bit 7 – The $\overline{IRQ}$ bit indicates the state of the $\overline{IRQ}$ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the $\overline{IRQ}$ output is low the $\overline{IRQ}$ bit will be high to indicate the interrupt or service request status. $\overline{IRQ}$ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.

ORDERING INFORMATION

EF68A50 | C | M | B/B

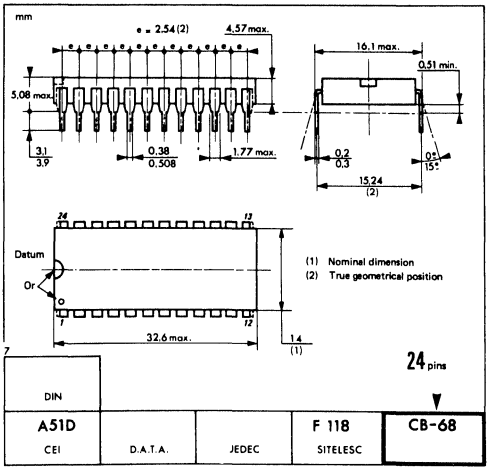
- EF68A50: Device
- C: Package
- M: Oper. temp.
- B/B: Screening level

The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

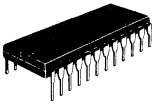
DEVICE	PACKAGE					OPER. TEMP			SCREENING LEVEL			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6850 (1.0 MHz)	●	●	●		●	●			●			
	●	●	●				●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68A50 (1.5 MHz)	●	●	●			●			●			
	●	●	●				●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68B50 (2.0 MHz)	●	●	●			●			●			
	●	●					●		●		●	

Examples : EF6850C, EF6850CV, EF6850CM, EF6850JM

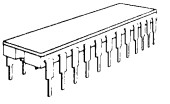
Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** −40°C to +85°C, **M :** −55°C to +125°C, *** :** may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.



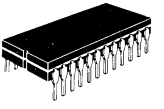
CB-68



P SUFFIX
PLASTIC PACKAGE



C SUFFIX
CERAMIC PACKAGE

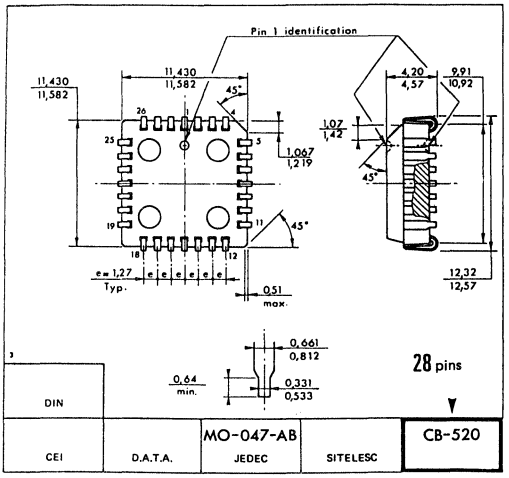


J SUFFIX
CERDIP PACKAGE

PHYSICAL DIMENSIONS

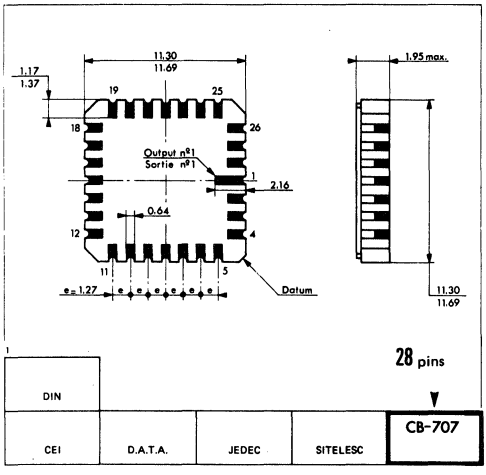
CB-520

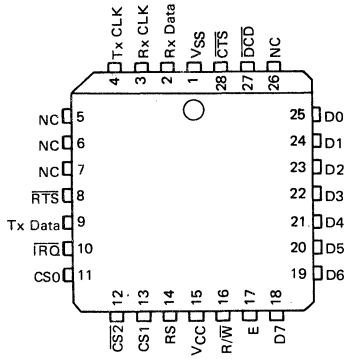
FN SUFFIX
PLCC 28



CB-707

E SUFFIX
LCCC 28





THOMSON COMPONENTS MOSTEK

EF6854

ADVANCED DATA-LINK CONTROLLER (ADLC)

The EF6854 ADLC performs the complex MPU/data communication link function for the "Advanced Data Communication Control Procedure" (ADCCP), High-Level Data-Link Control (HDLC) and Synchronous Data-Link Control (SDLC) standards. The ADLC provides key interface requirements with improved software efficiency. The ADLC is designed to provide the data communications interface for both primary and secondary stations in stand-alone, polling, and loop configurations.

- 6800 Compatible
- Protocol Features
  - Automatic Flag Detection and Synchronization
  - Zero Insertion and Deletion
  - Extendable Address, Control and Logical Control Fields (Optional)
  - Variable Word Length Information Field – 5-, 6-, 7-, or 8-Bits
  - Automatic Frame Check Sequence Generation and Check
  - Abort Detection and Transmision
  - Idle Detection and Transmission
- Loop Mode Operation
- Loop Back Self-Test Mode
- NRZ/NRZI Modes
- Quad Data Buffers for Each Rx and Tx
- Prioritized Status Register (Optional)
- MODEM/DMA/Loop Interface
- Three available versions : EF6854 (1.0 MHz)
  EF68A54 (1.5 MHz)
  EF68B54 (2.0 MHz)

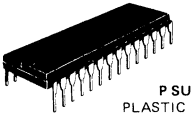
MOS

(N-CHANNEL, SILICON GATE)

ADVANCED DATA-LINK CONTROLLER

CASES

CB-132



P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

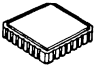
J SUFFIX
CERDIP PACKAGE

C SUFFIX
CERAMIC PACKAGE

CB-520

FN SUFFIX
PLCC 28

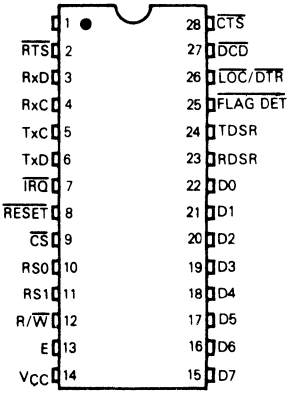
CB-707



E SUFFIX
LCCC 28

Hi-Rel versions available - See chapter 9

PIN ASSIGNMENT



MAXIMUM RATINGS

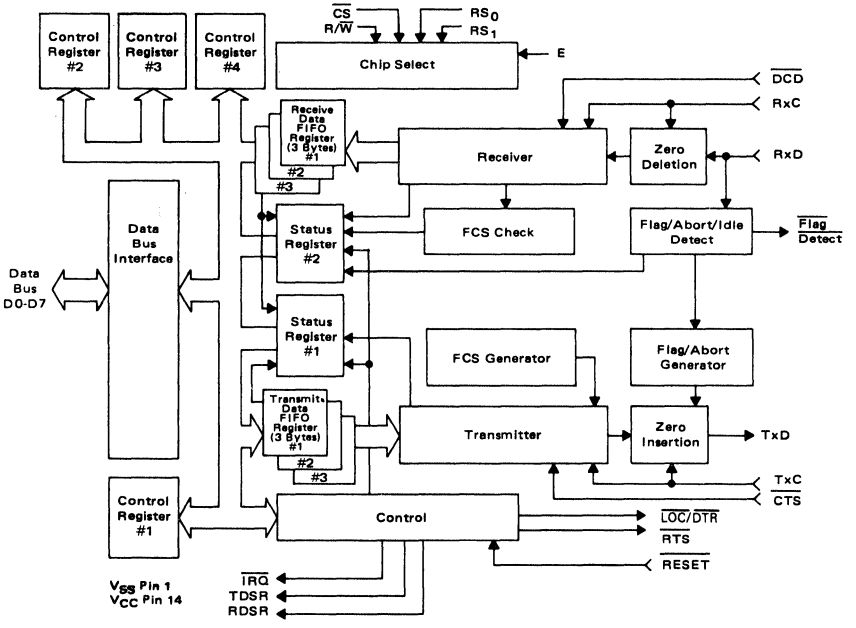
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range EF6854, EF68A54, EF68B54 EF6854, EF68A54, EF68B54 : V suffix EF6854, EF68A54 : M suffix	T_A	(T_L to T_H) 0 to 70 −40 to 85 −55 to +125	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHRACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Ceramic Cerdip PLCC	θ_{JA}	 115 60 65 100	°C/W

FIGURE 1 — ADLC GENERAL BLOCK DIAGRAM



POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts – Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	$V_{SS}+2.0$	–	–	V
Input Low Voltage		V_{IL}	–	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	All Inputs Except D0-D7	I_{in}	–	1.0	2.5	μA
Hi-Z (Off-State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = 5.25 V)	D0-D7	I_{IZ}	–	2.0	10	μA
dc Output High Voltage (I_{Load} = −205 μA)	D0-D7	V_{OH}	$V_{SS}+2.4$	–	–	V
(V_{Load} = −100 μA)	All Others		$V_{SS}+2.4$	–	–	
dc Output Low Voltage (I_{Load} = 1.6 mA))		V_{OL}	–	–	$V_{SS}+0.4$	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{OZ}	–	1.0	10	μA
Internal Power Dissipation (measured at T_A = 0°C)		P_{INT}	–	–	850*	mW
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	–	12.5	pF
	All Other Inputs		–	–	7.5	
	$\overline{IRQ}$	C_{out}	–	–	5.0	pF
	All Others		–	–	10	

*For temperatures below 0°C, P_{INT} will increase.

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	EF6854		EF68A54		EF68B54		Unit
		Min	Max	Min	Max	Min	Max	
Clock Pulse Width, Low (RxC, TxC)	PW_{CL}	700	–	450	–	280	–	ns
Clock Pulse Width, High (RxC, TxC)	PW_{CH}	700	–	450	–	280	–	ns
Serial Clock Frequency (RxC, TxC)	f_{SC}	–	0.66	–	1.0	–	1.5	MHz
Receive Data Setup Time	t_{RDSU}	150	–	100	–	50	–	ns
Receive Data Hold Time	t_{RDH}	60	–	60	–	60	–	ns
Request-to-Send Delay Time	t_{RTS}	–	680	–	460	–	340	ns
Clock-to-Data Delay for Transmitter	t_{TDD}	–	300	–	250	–	200	ns
Flag Detect Delay Time	t_{FD}	–	680	–	460	–	340	ns
$\overline{DTR}$ Delay Time	t_{DTR}	–	680	–	460	–	340	ns
Loop On-Line Control Delay Time	t_{LOC}	–	680	–	460	–	340	ns
RDSR Delay Time	t_{RDSR}	–	540	–	400	–	340	ns
TDSR Delay Time	t_{TDSR}	–	540	–	400	–	340	ns
Interrupt Request Release Time	t_{IR}	–	1.2	–	0.9	–	0.7	μs
$\overline{RESET}$ Pulse Width	t_{RESET}	1.0	–	0.65	–	0.40	–	μs
Input Rise and Fall Times (Except Enable) (0.8 V to 2.0 V)	t_r, t_f	–	1.0*	–	1.0*	–	1.0*	μs

*1.0 μs or 10% of the pulse width, whichever is smaller.

FIGURE 2 — BUS TIMING TEST LOADS

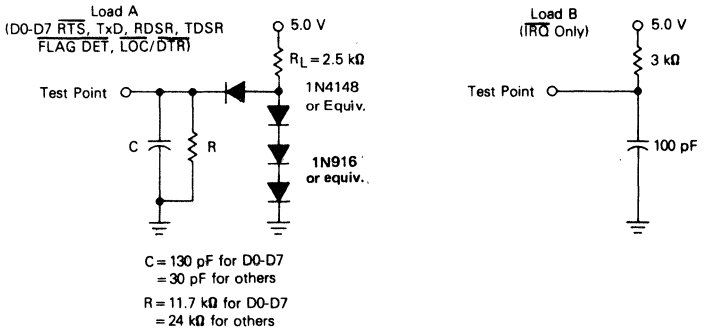


FIGURE 3 — RECEIVER DATA SETUP/HOLD, FLAG DETECT AND LOOP ON-LINE CONTROL DELAY TIMING

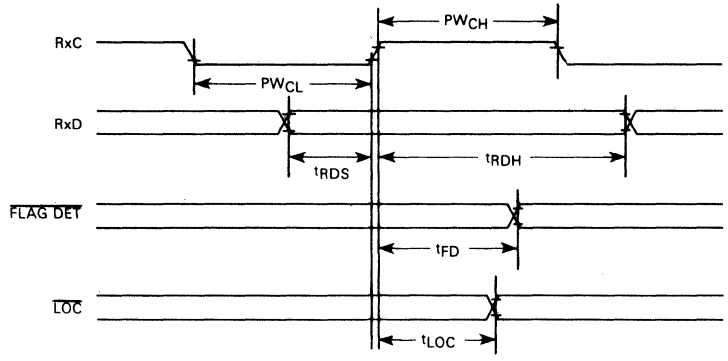
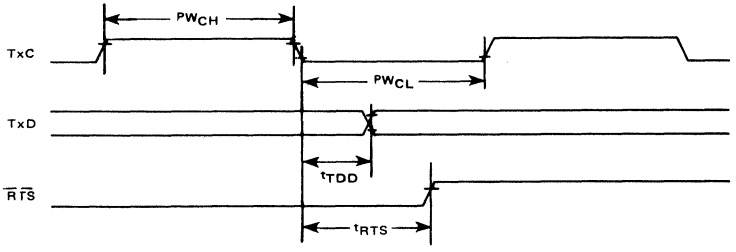
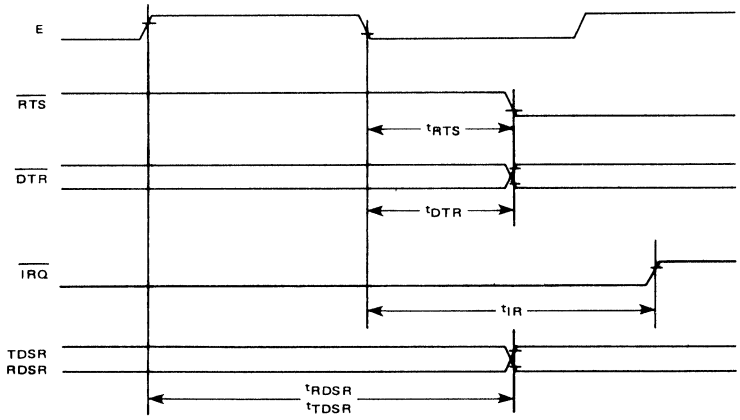


FIGURE 4 — TRANSMIT DATA OUTPUT DELAY AND REQUEST-TO-SEND DELAY TIMING



NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 5 — TDSR/RDSR DELAYS, $\overline{IRQ}$ RELEASE DELAY, $\overline{RTS}$ AND $\overline{DTR}$ DELAY TIMING



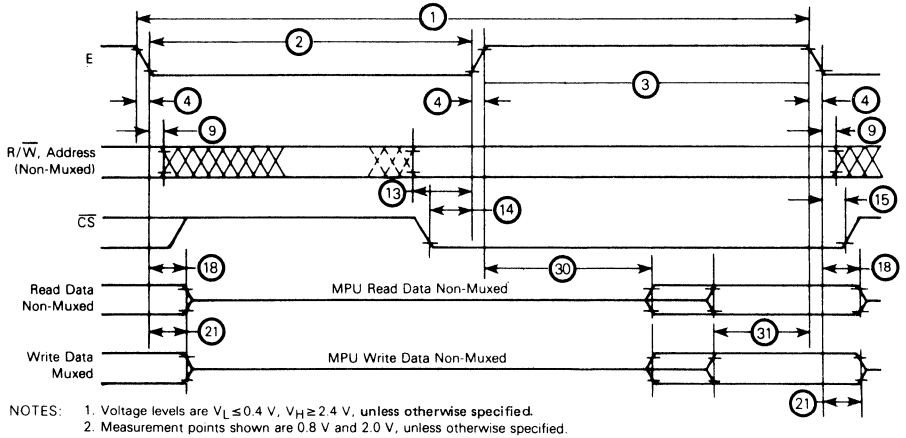
NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	EF6854		EF68A54		EF68B54		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

FIGURE 6 — BUS TIMING

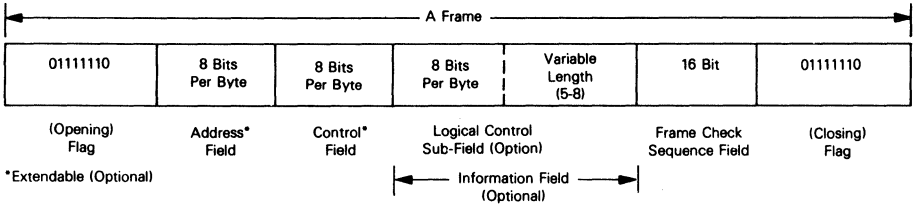


NOTES: 1. Voltage levels are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

FRAME FORMAT

The ADLC transmits and receives data (information or control) in a format called a frame. All frames start with an opening flag (F) and end with a closing flag (F). Between the opening flag and closing flag, a frame contains an address field, control field, information field (optional) and frame check sequence field.

FIGURE 7 – DATA FORMAT OF A FRAME



Flag (F) – The flag is the unique binary pattern (01111110). It provides the frame boundary and a reference for the position of each field of the frame.

The ADLC transmitter generates a flag pattern internally and the opening flag and closing flags are appended to a frame automatically. Two successive frames can share one flag for a closing flag of the first frame and for the opening flag of the next frame, if the "FF"/"F" control bit in the control register is reset.

The receiver searches for a flag on a bit-by-bit basis and recognizes a flag at any time. The receiver establishes the frame synchronization with every flag. The flags mark the frame boundary and reference for each field but they are not transferred to the Rx FIFO. The detection of a flag is indicated by the Flag Detect output and by a status bit in the status register.

Order of Bit Transmission – Address, control and information field bytes are transferred between the MPU and the ADLC in parallel by means of the data bus. The bit on D0 (data bus bit 0, pin 22) is serially transmitted first, and the first serially received bit is transferred to the MPU on D0. The FCS field is transmitted and received MSB first.

Address (A) Field – The 8 bits following the opening flag are the address (A) field. The A-field can be extendable if the Auto-Address Extend Mode is selected in control register #3. In the Address Extend Mode, the first bit (bit 0) in every address octet becomes the extend control bit. When the bit is "0", the ADLC assumes another address octet will follow, and when the bit is "1", the address extension is terminated. A "null" address (all "0's") does not extend. In the receiver, the Address Present status bit distinguishes the address field from other fields. When an address byte is available to be read in the receive FIFO register, the Address Present status bit is set and causes an interrupt (if enabled). The Address Present bit is set for every address octet when the Address Extend Mode is used.

Control (C) Field – The 8 bits following the address field is the control (link control) field. When the Extended Control Field bit in control register #3 is selected, the C-field is extended to 16 bits.

Information (I) Field – The I-field follows the C-field and precedes the FCS field. The I-field contains "data" to be transferred but is not always necessarily contained in every frame. The word length of the I-field can be selected from 5 to 8 bits per byte by control bits in control register #4. The I-field will continue until it is terminated by the FCS and closing flag. The receiver has the capability to handle a "partial" last byte. The last information byte can be any word length between 1 and 8 bits. If the last byte in the I-field is less than the selected word length, the receiver will right justify the received bits, fill the remaining bits of the receiver shift register with zeros, and transfer a full byte to the Rx FIFO. Regardless of selected byte length, the ADLC will transfer 8 bits of data to the data bus. Unused bits for word lengths of 5, 6, and 7 will be zeroed.

Logical Control (LC) Field – When the Logical Control Field Select bit, in control register #3, is selected the ADLC separates the I-field into two sub-fields. The first sub-field is the Logical Control field and the following sub-field is the "data" portion of the I-field. The logical control field is 8 bits and follows the C-field, which is extendable by octets, if it is selected. The last bit (bit 7) is the extend control bit, and if it is a "1", the LC-field is extended one octet.

NOTE

Hereafter the word "Information field" or "I-field" is used as the data portion of the information field, and excludes the logical control field. This is done in order to keep the consistency of the meaning of "Information field" as specified in SDLC, HDLC, and ADCCP standards.

Frame Check Sequence (FCS) Field – The 16 bits preceding the closing flag is the FCS field. The FCS is the "cyclic redundancy check character (CRCC)." The polynomial $x^{16}+x^{12}+x^5+1$ is used both for the transmitter and receiver. Both the transmitter and receiver polynomial registers are initialized to all "1's" prior to calculation of the FCS. The transmitter calculates the FCS on all bits of the address, control, logical control (if selected), and information fields, and transmits the complement of the resulting remainder as FCS. The receiver performs the similar computation on all bits of the address, control, logical control (if selected), information, and received FCS fields and compares the result to F0B8 (Hexadecimal). When the result matches F0B8, the Frame Valid status bit is set in the status register. If the result does not match, the Error status bit is set. The FCS generation, transmission, and checking are performed automatically by the ADLC transmitter and receiver. The FCS field is not transferred to the Rx FIFO.

Invalid Frame – Any valid frames should have at least the A-field, C-field, and FCS field between the opening flag and the closing flag. When invalid frames are received, the ADLC handles them as follows:

1. A short frame which has less than 25 bits between flags – the ADLC ignores the short frame and its reception is not reported to the MPU.
2. A frame less than 32 bits between the flags, or a frame 32 bits or more with an extended A-field or C-field that is not completed. – This frame is transferred into the Rx FIFO. The FCS/IF Error status bit indicates the reception of the invalid frame at the end of the frame.
3. Aborted Frame – The frame which is aborted by receiving an abort or DCD failure is also an invalid frame. Refer to "Abort" and "DCD status bit".

Zero Insertion and Zero Deletion – The Zero insertion and deletion, which allows the content of the frame to be transparent, are performed by the ADLC automatically. A binary 0 is inserted by the transmitter after any succession of five "1's" within a frame (A, C, LC, I, and FCS field). The receiver deletes a binary 0 that follows successive five continuous "1's" within a frame.

Abort – The function of prematurely terminating a data link is called "abort." The transmitter aborts a frame by sending at least eight consecutive "1's" immediately after the Tx Abort control bit in control register #4 is set to a "1". (Tx FIFO is also cleared by the Tx Abort control bit at the same time.) The abort can be extended up to (at least) 16 consecutive "1's", if the Abort Extend control bit in the control register #4 is set when an abort is sent. This feature is useful to force mark idle transmission. Reception of seven or more consecutive "1's" is interpreted as an abort by the receiver. The receiver responds to a received abort as follows:

1. An abort in an "out of frame" condition – an abort during the idle or time fill has no meaning. The abort reception is indicated in the status register as long as the abort condition continues; but neither an interrupt nor a stored condition occurs. The abort indication disappears after 15 or more consecutive "1's" are received (Received Idle status is set.)
2. An abort "in frame" after less than 26 bits are received after an opening flag – under this condition, any field of the aborted frame has not transferred to the MPU yet. The ADLC clears the aborted frame data in the FIFO and clears flag synchronization. Neither an interrupt nor a stored status occurs. The status indication is the same as (1) above.
3. An abort "in frame" after 26 bits or more are received after an opening flag – under this condition, some fields of the aborted frame might have been transferred onto the data bus. The abort status is stored in the receiver status register and the data of the aborted frame in the ADLC is cleared. The synchronization is also cleared.

Idle and Time Fill – When the transmitter is in an "out of frame" condition (the transmitter is not transmitting a frame), it is in an idle state. Either a series of contiguous flags (time fill) or a mark idle (consecutive "1's" on a bit-by-bit basis) is selected for the transmission in an idle state by the Flag/Mark Idle control bit. When the receiver receives 15 or more consecutive "1's", the Receive Idle status bit is set and causes an interrupt. The flags and mark idle are not transferred to the Rx FIFO.

OPERATION

INITIALIZATION

During a power-on sequence, the ADLC is reset via the $\overline{\text{RESET}}$ input and internally latched in a reset condition to prevent erroneous output transitions. The four control registers must be programmed prior to the release of the reset condition. The release of the reset condition is performed via software by writing a "0" into the Rx RS control bit (receiver) and/or Tx RS control bit (transmitter). The release of the reset condition must be done after the $\overline{\text{RESET}}$ input has gone high.

At any time during operation, writing a "1" into the Rx RS control bit or Tx RS control bit causes the reset condition of the receiver or the transmitter.

TRANSMITTER OPERATION

The Tx FIFO register cannot be pre-loaded when the transmitter is in a reset state. After the reset release, the Flag/Mark Idle control bit selects either the mark idle state (inactive idle) or the Flag "time fill" (active idle) state. This active or inactive mark idle state will continue until data is loaded into the Tx FIFO.

The availability of the Tx FIFO is indicated by the TDRA status bit under the control of the 2-Byte/1-Byte control bit. TDRA status is inhibited by the Tx RS bit or $\overline{\text{CTS}}$ input being high. When the 1-Byte mode is selected, one byte of the FIFO is available for data transfer when TDRA goes high. When the 2-Byte mode is selected, two successive bytes can be transferred when TDRA goes high.

The first byte (Address field) should be written into the Tx FIFO at the "Frame Continue" address. Then the transmission of a frame automatically starts. If the transmitter is in a mark idle state, the transfer of an address causes an opening flag within two or three transmitter clock cycles. If the transmitter has been in a time fill state, the current time fill flag being transmitted is assumed as an opening flag and the address field will follow it.

FIGURE 8a — ADLC TRANSMITTER STATE DIAGRAM
(C_ib_j refers to control register bit)

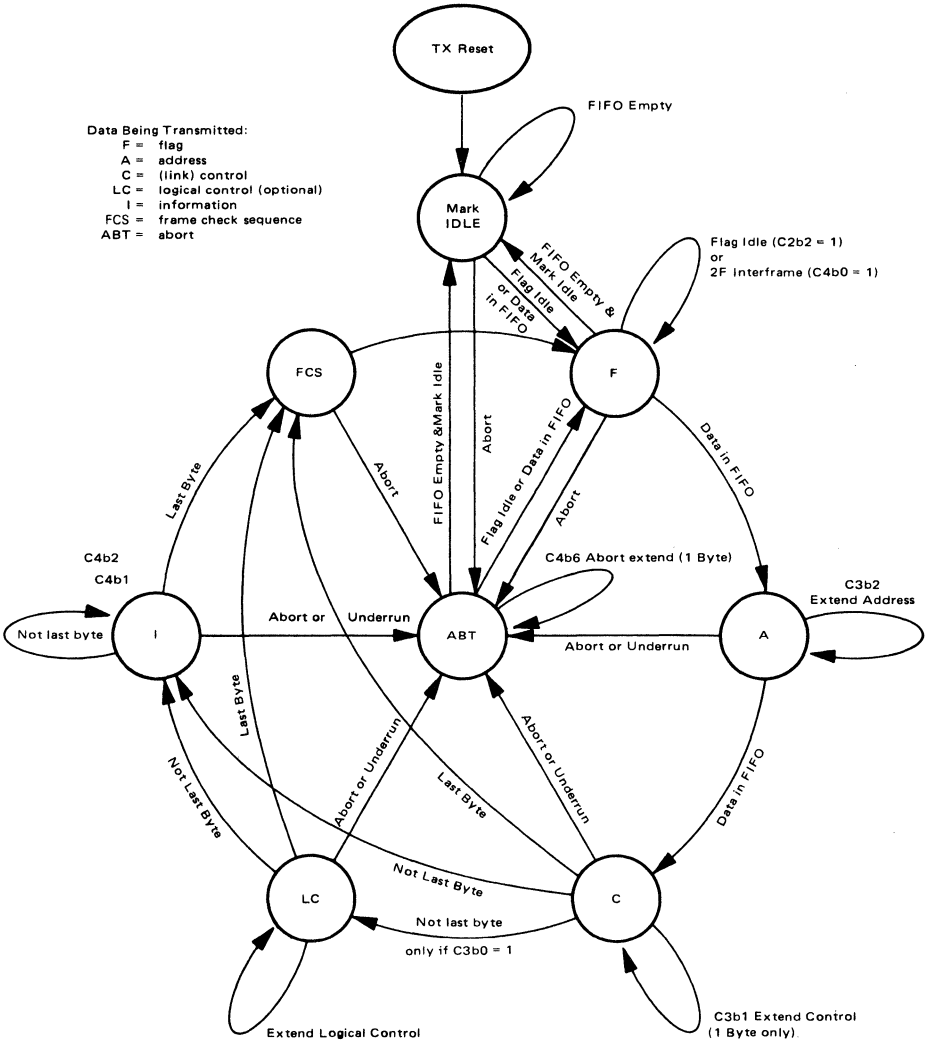
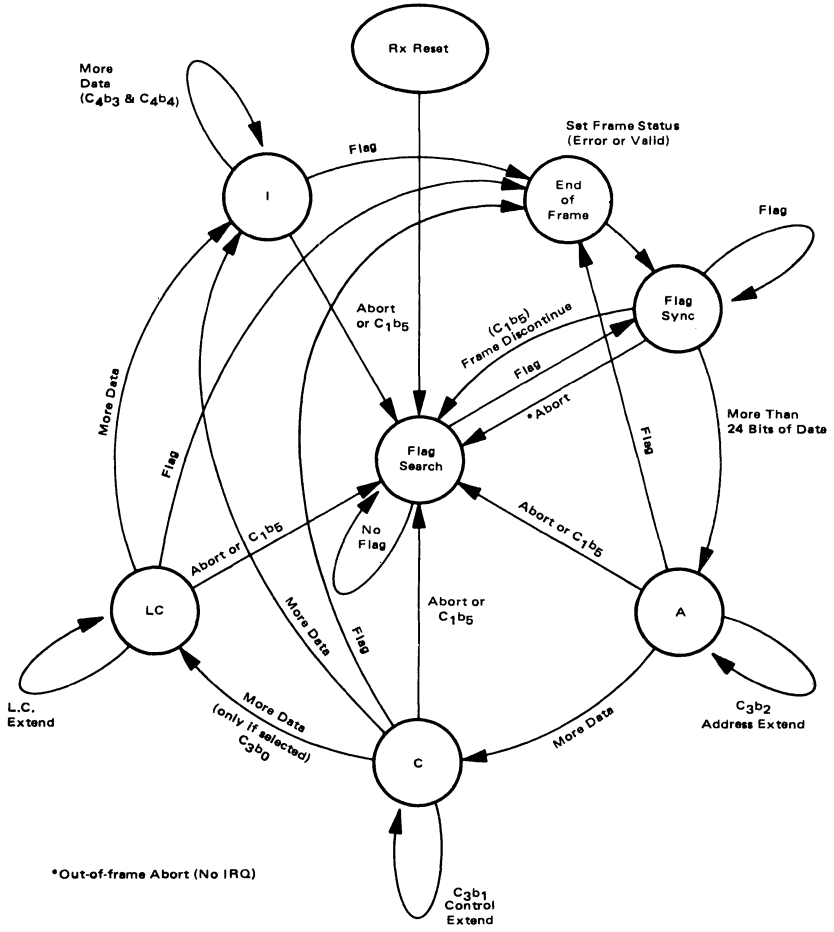


FIGURE 8b — ADLC RECEIVER STATE DIAGRAM



A frame continues as long as data is written into the Tx FIFO at the "Frame Continue" address. The ADLC internally keeps track of the field sequence in the frame. The frame format is described in the "FRAME FORMAT" section.

The frame is terminated by one of two methods. The most efficient way to terminate the frames from a software standpoint is to write the last data character into the Transmit FIFO "Frame Terminate" address (RS1, RS0=11) rather than the Transmit FIFO "Frame Continue" address (RS1, RS0=10). An alternate method is to follow the last write of data in the Tx FIFO "Frame Continue" address with the setting of the Transmit Last Data control bit. Either method causes the last character to be transmitted and the FCS field to automatically be appended along with a closing flag. Data for a new frame can be loaded into the Tx FIFO immediately after the old frame data, if TDRA is high. The closing Flag can serve as the opening Flag of the next frame or separate opening and closing Flags may be transmitted. If a new frame is not ready to be transmitted, the ADLC will automatically transmit the Active (Flag) or Inactive (Mark) Idle condition.

If the Tx FIFO becomes empty at any time during frame transmission (the FIFO has no data to transfer into transmitter shift register during transmission of the last half of the

next to last bit of a word), an underrun will occur and the transmitter automatically terminates the frame by transmitting an abort. The underrun state is indicated by the Tx Underrun status bit.

Any time the Tx ABORT Control bit is set, the transmitter immediately aborts the frame (transmits at least 8 consecutive "1's") and clears the Tx FIFO. If the Abort Extend Control bit is set at the time, an idle (at least 16 consecutive "1's") is transmitted. An abort or idle in an "out of frame" condition can be useful to gain 8 or 16 bits of delay. (For an example, see "Programming Considerations.")

The $\overline{CTS}$ (Clear-to-Send) input and $\overline{RTS}$ (Request-to-Send) output are provided for a MODEM or other hardware interface.

The TDRA/FC status bit (when selected to be Frame Complete Status) can cause an interrupt upon frame completion (i.e., a flag or abort completion).

Details regarding the inputs and outputs, status bits, control bits, and FIFO operation are described in their respective sections.

RECEIVER OPERATION

Data and a pre-synchronized clock are provided to the ADLC receiver section by means of the Receive Data (RxD) and Receive Clock (RxC) inputs. The data is a continuous stream of binary bits with the characteristic that a maximum of five "1's" can occur in succession unless Abort, Flag, or Idling condition occurs. The receiver continuously (on a bit-by-bit basis) searches for Flags and Aborts.

When a flag is detected, the receiver establishes frame synchronization to the flag timing. If a series of flags is received, the receiver resynchronizes to each flag.

If the frame is terminated before the internal buffer time expires (the frame data is less than 25 bits after an opening flag), the frame is simply ignored. Noise on the data input (RxD) during time fill can cause this kind of invalid frame.

The received serial data enters a 32-bit shift register (clocked by RxC) before it is transferred into the Rx Data FIFO. Synchronization is established when a Flag is detected in the first eight locations of the shift register. Once synchronization has been achieved, data is clocked through to the last byte location of the shift register where it is transferred byte-per-byte into the Rx Data FIFO. The Rx Data FIFO is clocked by E to cause received data to move through the FIFO to the last empty register location. The Receiver Data Available status bit (RDA) indicates when data is present in the last register (Register #3) for the 1-Byte Transfer Mode. The 2-Byte Transfer Mode causes the RDA status bit to indicate data is available when the last two FIFO register locations (Registers #2 and #3) are full. If the data character present in the FIFO is an address octet, the status register will exhibit an Address Present status condition. Data being available in the Rx Data FIFO causes an interrupt to be initiated (assuming the receiver interrupt is enabled, RIE="1"). The MPU will read the ADLC Status Register as a result of the interrupt or in its turn in a polling sequence. RDA or Address Present will indicate that receiver data is available and the MPU should subsequently read the Rx Data FIFO register. The interrupt and status bit will then be reset automatically. If more than one character had been received and was resident in the Rx Data FIFO, subsequent E clocks will cause the FIFO to update and the RDA status bit and interrupt will again be SET. In the 2-Byte Transfer Mode both data bytes may be read on consecutive E cycles. Address Present provides for 1 byte transfers only.

The sequence of each field in the received frame is automatically handled by the ADLC. The frame format is described in the "FRAME FORMAT" section.

When a closing flag is received, the frame is terminated. The 16 bits preceding the closing flag are regarded as the FCS and are not transferred to the MPU. Whatever data is present in the most-significant byte portion of the receiver buffer register it is right justified and transferred to the Rx FIFO. The frame boundary pointer, which is explained in the "Rx FIFO REGISTER" section, is set simultaneously in the Rx FIFO. The frame boundary pointer sets the Frame Valid status bit (when the frame was completed with no error) or the FCS/IF Error Status bit (when the frame was completed with error) when the last byte of the frame appears at the last location of the Rx FIFO. As long as the Frame Valid or FCS/IF Error status bit is set, the data transfer from the second location of the Rx FIFO to the last location of the Rx FIFO is inhibited.

Any time the Frame Discontinue control bit is set, the ADLC discards the current frame data in the ADLC without dropping flag synchronization. This feature can be used to ignore a frame which is addressed to another station.

The reception of an abort or idle is explained in the "FRAME FORMAT" section. The details regarding the inputs, outputs, status bits, control bits, and Rx FIFO operation are described in their respective sections.

LOOP MODE OPERATION

The ADLC in the loop mode, not only performs the transmission and receiving of data frames in the manner previously described, but also has additional features for gaining and relinquishing loop control. In Figure 9a, a configuration is shown which depicts loop mode operation. The system configuration shows a primary station and several secondary stations. The loop is always under control of the primary station. When the primary wants to receive data, it transmits a Poll sequence and allows frame transmission to secondary stations on the loop. Each secondary is in series and adds one bit of delay to the loop. Secondary A in the figure receives data from the primary via its Rx Data Input, delays the data 1 bit, and transmits it to secondary B via its Tx Data Output. Secondaries B, C, and D operate in a similar manner. Therefore, data passes through each secondary and is received back by the primary controller.

Certain protocol rules must be followed in the manner by which the secondary station places itself on-loop (connects its transmitter output to the loop), goes active on the loop (starts transmitting its own station's data on the loop), and goes off the loop (disconnects its transmitter output). Otherwise loop data to other stations down loop would be interfered. The data stream always flows the same way and the order in which secondary terminals are serviced is determined by the hardware configuration. The primary controller times the delay through the loop. Should it exceed n+1 bit times, where n is the number of secondary terminals on the loop, it will indicate a loop failure. Control is transferred to a secondary by transmitting a "Go Ahead" signal following the closing Flag of a polling frame (request for a response from the secondary) from the primary station. The "Go Ahead" from the primary is a "0" and seven "1's" followed by mark

FIGURE 9a — TYPICAL LOOP CONFIGURATION

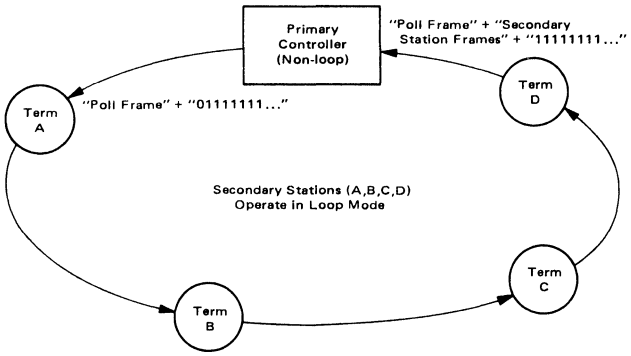
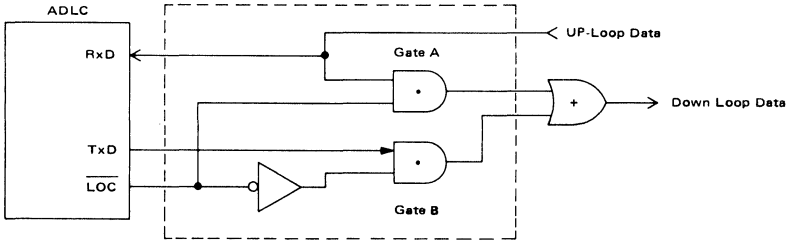


FIGURE 9b — EXAMPLE OF EXTERNAL LOOP LOGIC



idling. The primary can abort its response request by interrupting its idle with flags. The secondary should immediately stop transmission and return control back to the primary. When the secondary completes its frame, a closing flag is transmitted followed by all "1's". The primary detects the final 01111111...("Go Ahead" to the primary) and control is given back to the primary. Note that, if a down-loop secondary (e.g., station D) needs to insert information following an up-loop station (e.g., station A), the go ahead to station D is the last "0" of the closing flag from station A followed by "1's".

The ADLC in the primary station should operate in a non-loop full-duplex mode. The ADLC in the secondaries should operate in a loop mode, monitoring up-loop data on its receiver data input. The ADLC can recognize the necessary sequences in the data stream to automatically go on/off the loop and to insert its own station data. The procedure is the following and is summarized in Table 1.

(1) Go On-Loop — When the ADLC powers up, the terminal station will be off line. The first task is to become an active terminal on the loop. The ADLC must be connected to a Loop Link via an external switch as shown in Figure 9a. After a hardware reset, the ADLC $\overline{LOC}/\overline{DTR}$ Output will be in the high state and the up-loop receive data repeated through gate A to the down Loop stations. Any Up-Loop transmission will be received by the ADLC. The Loop Mode/Non-Loop Mode Control bit (bit 5 in Control Register 3) must be set to place the ADLC in the Loop Mode. The ADLC now monitors its Rx Data input for a string of seven consecutive "1's" which will allow a station to go on line. The Loop operation may be monitored by use of the Loop Status bit in Status Register 1. After power up and reset, this bit is a zero. When seven consecutive "1's" are received by the ADLC the $\overline{LOC}/\overline{DTR}$ output will go to a low level, disabling gate A (refer to Figure 9b), enabling gate B and connecting the ADLC Tx Data output to the down Loop stations. The up Loop data is now repeated to the down Loop stations via the ADLC. A 1-bit delay is inserted in the data (in NRZI mode, there will be a 2-bit delay) as it circulates through the ADLC. The ADLC is now on-line and the Loop Status bit in Status Register 1 will be at a one.

(2) Go Active after Poll — The receiver section will monitor the up-link data for a general or addressed poll command and the Tx FIFO should be loaded with data so that when the go ahead sequence of a zero followed by seven "1's" (01111111---) is detected, transmission can be initiated immediately. When the polling frame is detected, the Go-Active-On-Poll control bit must be set (bit 6 in Control

TABLE 1 — SUMMARY OF LOOP MODE OPERATION

STATE	RX SECTION	TX SECTION	LOOP STATUS BIT
OFF-LOOP	Rx section receives data from loop and searches for 7 "1's" (when On-Loop Control bit set) to go ON-LOOP.	Inactive 1) NRZ MODE. Tx data output is maintained "high" (mark). 2) NRZI MODE. Tx data output reflects the Rx data input state delayed by one bit time. (Not normally connected to loop.) The NRZI data is internally decoded to provide error-free transitions to On-Loop mode.	"0"
ON-LOOP	1) When Go-Active on poll bit is set, Rx section searches for 01111111 pattern (the EOP or 'Go Ahead') to become the active terminal on the loop. 2) When On-Loop control bit is reset, Rx section searches for 8 "1's" to go OFF-Loop.	Inactive 1) NRZ MODE. Tx data output reflects Rx data input state delayed one bit time. 2) NRZI MODE. Tx data output reflects Rx data input state delayed 2 bit times.	"1"
ACTIVE	Rx section searches for flag (an interrupt from the loop controller) at Rx data input. Received flag causes $\overline{FD}$ output to go low. IRQ is generated if $\overline{RIE}$ and FDSE control bits are set.	Tx data originates within ADLC until Go Active on Poll bit is reset and a flag or Abort is completed. Then returns to ON-Loop state.	"0"

Register 3). A maximum of seven bit times are available to set this control bit after the closing flag of the poll. When the Go-Ahead is detected by the receiver, the ADLC will automatically change the seventh one to a zero so that the repeated sequence out gate B in Figure 9b is now an opening flag sequence (011111110). Transmission now continues from the Tx FIFO with data (address, control, etc.) as previously described. When the ADLC has gone active-on-poll, the Loop Status bit in Status Register 1 will go to a zero. The receiver searches for a flag, which indicates that the primary station is interrupting the current operation.

(3) Go Inactive when On-Loop — The Go-Active-On-Poll control bit may be $\overline{RESET}$ at any time during transmission. When the frame is complete (the closing Flag or abort is transmitted), the Loop is automatically released and the station reverts back to being just a 1-bit delay in the Loop, repeating up-link data. If the Go-Active-On-Poll control bit is not reset by software and the final frame is transmitted (Flag/Mark Idle bit = 0), then the transmitter will mark idle and will not release the loop to up-loop data. A Tx Abort command would have to be used in this case in order to go inactive when on the loop. Also, if the Tx FIFO was not preloaded with data (address, control, etc.) prior to changing the "Go Ahead Character" to a Flag, the ADLC will either transmit flags (active idle character) until data is loaded (when Flag/Mark Idle Control bit is high) or will go into an underrun condition and transmit an Abort (when Flag/Mark Idle control bit is low). When an abort is transmitted, the Go-Active-on-Poll control bit is reset automatically and the ADLC reverts to its repeating mode, (TxD = delayed RxD). When the ADLC transmitter lets go of the loop, the Loop Status bit will return to a "1", indicating normal on-loop retransmission of up-loop data.

(4) Go Off-Loop — The ADLC can drop off the Loop (go off-line) similar to the way it went on-line. When the Loop On-Line control bit is reset the ADLC receiver section looks for eight successive "1's" before allowing the $\overline{LOC}/\overline{DTR}$ output to return high (the inactive state). Gate A in Figure 9b will be enabled and gate B disabled allowing the loop to maintain continuity without disturbance. The Loop Status bit will show an off-line condition (logical zero).

SIGNAL DESCRIPTIONS

All inputs of ADLC are high-impedance and TTL-compatible level inputs. All outputs of the ADLC are compatible with standard TTL. Interrupt Request ($\overline{IRQ}$), however, is an open-drain output (no internal pullup).

INTERFACE FOR MPU

Bidirectional Data Bus (D0-D7) — These data bus I/O ports allow the data transfer between ADLC and system bus. The data bus drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ADLC read operation.

Enable Clock (E) — E activates the address inputs ($\overline{CS}$, RS0, and RS1) and R/$\overline{W}$ input and enables the data transfer on the data bus. E also moves data through the Tx FIFO and Rx FIFO. E should be a free-running clock such as the EF6800 MPU system clock.

Chip Select ($\overline{CS}$) — An ADLC read or write operation is enabled only when the $\overline{CS}$ input is low and the E clock input is high. (E•$\overline{CS}$).

Register Selects (RS0, RS1) – When the Register Select inputs are enabled by (E•$\overline{CS}$), they select internal registers in conjunction with the Read/Write input and Address Control bit (control register 1, bit 0). Register addressing is defined in Table 2.

Read/Write Control Line (R/$\overline{W}$) – The R/$\overline{W}$ input controls the direction of data flow on the data bus when it is enabled by (E•$\overline{CS}$). When R/$\overline{W}$ is high, the I/O Buffer acts as an output driver and as an input buffer when low. It also selects the Read Only and Write Only registers within the ADLC.

Reset Input ($\overline{RESET}$) – The $\overline{RESET}$ input provides a means of resetting the ADLC from a hardware source. In the "low state," the $\overline{RESET}$ input causes the following:

- Rx Reset and Tx Reset are SET causing both the Receiver and Transmitter sections to be held in a reset condition.
- Resets the following control bits: Transmit Abort, $\overline{RTS}$, Loop Mode, and Loop On-Line/$\overline{DTR}$.
- Clears all stored status condition of the status registers.
- Outputs: $\overline{RTS}$ and $\overline{LOC/DTR}$ go high. TxD goes to the mark state ("1's" are transmitted).

When RESET returns "high" (the inactive state) the transmitter and receiver sections will remain in the reset state until Tx Reset and Rx Reset are cleared via the data bus under software control. The Control Register bits affected by $\overline{RESET}$ cannot be changed when $\overline{RESET}$ is "low."

Interrupt Request Output ($\overline{IRQ}$) – $\overline{IRQ}$ will be low if an interrupt situation exists and the appropriate interrupt enable has been set. The interrupt remains as long as the cause for the interrupt is present and the enable is set. $\overline{IRQ}$ will be low as long as the $\overline{IRQ}$ status bit is set and is high if the $\overline{IRQ}$ status bit is not set.

CLOCK AND DATA OF TRANSMITTER AND RECEIVER

Transmitter Clock Input (TxC) – The transmitter shifts data on the negative transition of the TxC clock input. When the Loop Mode or Test Mode is selected, TxC should be the same frequency and phase as the RxC clock. The data rate of the transmitter should not exceed the E frequency.

Receiver Clock Input (RxC) – The receiver samples the data on the positive transition of the RxC clock. RxC should be synchronized with receive data externally.

Transmit Data Output (TxD) – The serial data from the transmitter is coded in NRZ or NRZI (Zero Complement) data format.

Receiver Data Input (RxD) – The serial data to be received by the ADLC can be coded in NRZ or NRZI (Zero Complement) data format. The data rate of the receiver should not exceed the E frequency. If a partial byte reception is possible at the end of a frame, the maximum data rate of the receiver is indicated by the following relationship:

$$f_{RxC} \leq \frac{1}{2t_E + 300\ ns}$$

where t_E is the period of E.

PERIPHERAL/MODEM CONTROL

Request-to-Send Output ($\overline{RTS}$) – The Request-to-Send output is controlled by the Request-to-Send control bit in conjunction with the state of the transmitter section. When the $\overline{RTS}$ bit goes high, the $\overline{RTS}$ output is forced low. When the $\overline{RTS}$ bit returns low, the $\overline{RTS}$ output remains low until the end of the frame and there is no further data in the Tx FIFO for a new frame. The positive transition of $\overline{RTS}$ occurs after the completion of a Flag, an Abort, or when the RTS control bit is reset during a mark idling state. When the $\overline{RESET}$ input is low, the $\overline{RTS}$ output goes high.

Clear-to-Send Input ($\overline{CTS}$) – The $\overline{CTS}$ input provides a real-time inhibit to the TDRA status bit and its associated interrupt. The positive transition of $\overline{CTS}$ is stored within the ADLC to ensure its occurrence will be acknowledged by the system. The stored $\overline{CTS}$ information and its associated $\overline{IRQ}$ (if enabled) are cleared by writing a "1" in the Clear Tx Status bit or in the Transmitter Reset bit.

Data-Carrier-Detect Inupt ($\overline{DCD}$) – The $\overline{DCD}$ input provides a real-time inhibit to the receiver section. A high level on the $\overline{DCD}$ input resets and inhibits the receiver register, but data in the Rx FIFO from a previous frame is not disturbed. The positive transition of $\overline{DCD}$ is stored within the ADLC to ensure that its occurrence will be acknowledged by the system. The stored $\overline{DCD}$ information and its associated $\overline{IRQ}$ (if enabled) are cleared by means of the Clear Rx Status Control bit or by the Rx Reset bit.

Loop On-Line Control/Data Terminal Ready Output ($\overline{LOC/DTR}$) – The $\overline{LOC/DTR}$ output serves as a $\overline{DTR}$ output in the non-loop mode or as a Loop Control output in the loop mode. When $\overline{LOC/DTR}$ output performs the $\overline{DTR}$ function, it is turned on and off by means of the $\overline{LOC/DTR}$ control bit. When the $\overline{LOC/DTR}$ control bit is high the $\overline{DTR}$ output will be low. In the loop mode the $\overline{LOC/DTR}$ output provides the means of controlling the external loop interface hardware to go On-line or Off-line. When the $\overline{LOC/DTR}$ control bit is SET and the loop has "idled" for 7 bit times or more (RxD = 01111111...), the $\overline{LOC/DTR}$ output will go low (on-line). The $\overline{RESET}$ input being low will cause the $\overline{LOC/DTR}$ output to be high.

Flag Detect Output ($\overline{FD}$) – An output to indicate the reception of a flag and initiate an external time-out counter for the loop mode operation. The $\overline{FD}$ output goes low for 1 bit time beginning at the last bit of the flag character, as sampled by the receiver clock (RxC).

DMA INTERFACE

Receiver Data Service Request Output (RDSR) – The RDSR Output is provided primarily for use in DMA Mode operation and indicates (when high) that the Rx FIFO requests service (RSDR output reflects the RDA status bit regardless of the state of the RDSR mode control bit in CR1). If the prioritized Status Mode is selected, RDSR will be inhibited when any other receiver status conditions are present. RDSR goes low when the Rx FIFO is read.

Transmitter Data Service Request Output (TDSR) – The TDSR Output is provided for DMA mode operation and indicates (when high) that the Tx FIFO request service regardless of the state of the TDSR Mode Control bit in CR1. TDSR goes low when the Tx FIFO is loaded. TDSR is inhibited by: the Tx RS control bit being SET, $\overline{\text{RESET}}$ being low, or CTS being high. If the prioritized status mode is used, Tx Underrun also inhibits TDSR. TDSR reflects the TDRA status bit except in the FC mode. In the FC mode the TDSR line is inhibited.

ADLC REGISTERS

Eight registers in the ADLC can be accessed by means of the MPU data and address buses. The registers are defined as read-only or write-only according to the direction of information flow. The addresses of these registers are defined in Table 2. The transitter FIFO register can be accessed by two different addresses, the "Frame Terminate" address and the "Frame Continue" address. (The function of these addresses are discussed in the FIFO section.)

TABLE 2 – REGISTER ADDRESSING

Register Selected	R/$\overline{W}$	RS1	RS0	Address Control Bit (C_1b_0)
Write Control Register #1	0	0	0	X
Write Control Register #2	0	0	1	0
Write Control Register #3	0	0	1	1
Write Transmit FIFO (Frame Continue)	0	1	0	X
Write Transmit FIFO (Frame Terminate)	0	1	1	0
Write Control Register #4	0	1	1	1
Read Status Register #1	1	0	0	X
Read Status Register #2	1	0	1	X
Read Receiver FIFO	1	1	X	X

RECEIVER DATA FIRST-IN FIRST-OUT REGISTER

Rx FIFO – The Rx FIFO consists of three 8-bit registers which are used for the buffer storage of received data. Data bytes are always transferred from a full register to an adjacent empty register; and both phases of the E input clock are used for the data transfer. Each register has pointer bits which point the frame boundary. When these pointers appear at the last FIFO location, they update the Address Present, Frame Valid, or FCS/IF Error status bits.

The RDA status bit indicates the state of the Rx FIFO. When RDA status bit is "1", the Rx FIFO is ready to be read. The RDA status is controlled by the 2-Byte/1-Byte control bit. When overrun occurs, the data in the first byte of the Rx FIFO are not longer valid.

Both the Rx Reset bit and $\overline{\text{RESET}}$ input clear the Rx FIFO. Abort ("in Frame") and a high level on the $\overline{\text{DCD}}$ input also clears the Rx FIFO, but the last bytes of the previous frame, which are separated by the frame boundary pointer, are not disturbed.

TRANSMITTER DATA FIRST-IN FIRST-OUT REGISTER

Tx FIFO – The Tx FIFO consists of three 8-bit registers which are used for buffer storage of data to be transmitted. Data is always transferred from a full register to an empty adjacent register; the transfer occurs on both phases of the E input clock. The Tx FIFO can be addressed by two different register addresses, the "Frame Continue" address and the "Frame Terminate" address. Each register has pointer bits which point to the frame boundary. When a data byte is written at the "Frame Continue" address, the pointer of the first FIFO register is set. When a data byte is written at the "Frame Terminate" address, the pointer of the first FIFO register is reset. Rx RS control bit or Tx Abort control bit resets all pointers. The pointer will shift through the FIFO. When a positive transition is detected at the third location of FIFO, the transmitter initiates a frame with an open flag. When the negative transition is detected at the third location of FIFO, the transmitter closes a frame, appending the FCS and closing Flag to the last byte.

The Tx last control bit can be used instead of using the "Frame Terminate" address. When the Tx last control bit is set with a "1", the logic searches the last byte location in the FIFO and resets the pointer in the FIFO register.

The status of Tx FIFO is indicated by the TDRA status bit. When TDRA is "1", the Tx FIFO is available for loading data. The TDRA status is controlled by the 2-Byte/1-Byte control bit. The Tx FIFO is reset by both Tx Reset and $\overline{\text{RESET}}$ input. During this reset condition or when $\overline{\text{CTS}}$ input is high, the TDRA status bit is suppressed and data loading is inhibited.

ADLC INTERNAL REGISTER STRUCTURE

		RS1 RS0 = 00	RS1 RS0 = 01	RS1 RS0 = 10	RS1 RS0 = 11
	Bit #	Status Register #1	Status Register #2	Receiver Data Register	
Read Only Registers	0	RDA	Address Present	Bit 0	Same as RS1, RS0 = 10
	1	Status #2 Read Request	Frame Valid	Bit 1	
	2	Loop	Inactive Idle Received	Bit 2	
	3	Flag Detected (When Enabled)	Abort Received	Bit 3	
	4	$\overline{CTS}$	FCS Error	Bit 4	
	5	Tx Underrun	$\overline{DCD}$	Bit 5	
	6	TDRA/Frame Complete	Rx Overrun	Bit 6	
	7	IRQ Present	RDA (Receiver Data Available)	Bit 7	

					Transmitter Data	Transmitter Data	
	Bit #	Control Register #1	Control Register #2 ($C_1b_0 = 0$)	Control Register #3 ($C_1b_0 = 1$)	(Continue Data)	(Last Data) ($C_1b_0 = 0$)	Control Register #4 ($C_1b_0 = 1$)
Write Only Registers	0	Address Control (AC)	Prioritized Status Enable	Logical Control Field Select	Bit 0	Bit 0	Double Flag/Single Flag Interframe Control
	1	Receiver Interrupt Enable (RIE)	2 Byte/1 Byte Transfer	Extended Control Field Select	Bit 1	Bit 1	Word Length Select Transmit #1
	2	Transmitter Interrupt Enable (TIE)	Flag/Mark Idle	Auto, Address Extension Mode	Bit 2	Bit 2	Word Length Select Transmit #2
	3	RDSR Mode (DMA)	Frame Complete/ TDRA Select	01/11 Idle	Bit 3	Bit 3	Word Length Select Receive #1
	4	TDSR Mode (DMA)	Transmit Last Data	Flag Detected Status Enable	Bit 4	Bit 4	Word Length Select Receive #2
	5	Rx Frame Discontinue	CLR Rx Status	Loop/Non-Loop Mode	Bit 5	Bit 5	Transmit Abort
	6	Rx RESET	CLR Tx Status	Go Active on Poll/Test	Bit 6	Bit 6	Abort Extend
	7	Tx RESET	RTS Control	Loop On-Line Control DTR	Bit 7	Bit 7	NRZI/NRZ

CONTROL REGISTER 1 (CR1)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
0	0	0	X	TxRS	RxRS	Discontinue	TDSR Mode	RDSR Mode	TIE	RIE	AC

b0 — Address Control (AC) — AC provides another RS (Register Select) signal internally. The AC bit is used in conjunction with RS0, RS1, and R/$\overline{W}$ inputs to select particular registers, as shown in Table 2.

b1 — Receiver Interrupt Enable (RIE) — RIE enables/disables the interrupt request caused by the receiver section. 1...enable, 0...disable.

b2 — Transmitter Interrupt Enable (TIE) — TIE enables/disables the interrupt request caused by the transmitter. 1...enable, 0...disable.

b3 — Receiver Data Service Request Mode (RDSR MODE) — The RDSR MODE bit provides the capability of operation with a bus system in the DMA mode when used in conjunction with the prioritized status mode. When RDSR MODE is set, an interrupt request caused by RDA status is inhibited, and the ADLC does not request data transfer via the $\overline{IRQ}$ output.

b4 — Transmitter Data Service Request Mode (TDSR MODE) — The TDSR MODE bit provides the capability of operation with a bus system in the DMA mode when used in conjunction with the prioritized status mode. When TDSR MODE is set, an interrupt request caused by TDRA status is inhibited, and the ADLC does not request a data transfer via the $\overline{IRQ}$ output.

b5 — Rx Frame Discontinue (DISCONTINUE) — When the DISCONTINUE bit is set, the currently received frame is ignored and the ADLC discards the data of the current frame. The DISCONTINUE bit only discontinues the currently received frame and has no affect on subsequent frames, even if a following frame has entered the receiver section. The DISCONTINUE bit is automatically reset when the last byte of the frame is discarded. When the ignored frame is aborted by receiving an Abort or DCD failure, the DISCONTINUE bit is also reset.

b6 — Receiver Reset (Rx RS) — When the Rx RS bit is "1", the receiver section stays in the reset condition. All receiver sections, including the Rx FIFO register and the receiver status bits in both status registers, are reset. (During reset, the stored DCD status is reset but the DCD status bit follows the $\overline{DCD}$ input.) Rx RS is set by forcing a low level on the $\overline{RESET}$ input or by writing a "1" into the bit from the data bus. Rx RS must be reset by writing a "0" from the data bus after $\overline{RESET}$ has gone high.

b7 — Transmitter Reset (Tx RS) — When the Tx RS bit is "1", the transmitter section stays in the reset condition and transmits marks ("1's"). All transmitter sections, including the Tx FIFO and the transmitter status bits, are reset (FIFO cannot be loaded). During reset, the stored CTS status is reset but the CTS status bit follows the $\overline{CTS}$ input. Tx RS is set by forcing a low level on the $\overline{RESET}$ input or by writing a "1" from the data bus. It must be reset by writing a "0" after $\overline{RESET}$ has gone high.

CONTROL REGISTER 2 (CR2)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
0	1	0	0	RTS	CLR TxST	CLR RxST	Tx Last	FC/TDRA Select	F/M Idle	2/1 Byte	PSE

b0 — Prioritized Status Enable (PSE) — When the PSE bit is SET, the status bits in both status registers are prioritized as defined in the Status Register section. When PSE is low, the status bits indicate current status without bit suppression by other status bits. The exception to this rule is the $\overline{CTS}$ status bit which always supresses the TDRA status.

b1 — 2-Byte/1-Byte Transfer (2/1 Byte) — When the 2/1 Byte bit is RESET the TDRA and RDA status bits then will indicate the availability of their respective data FIFO registers for a single-byte data transfer. Similarly, if 2/1 Byte is set, the TDRA and RDA status bit indicate when two bytes of data can be moved without a second status read.

b2 — Flag/Mark Idle Select (F/M Idle) — The F/M Idle bit selects Flag characters or bit-by-bit Mark Idle for the time fill or the idle state of the transmitter. When Mark Idle is selected, Go-Ahead code can be generated for loop operation in conjunction with the 01/11 Idle control bit (C_3b_3). 1...Flag time fill, 0...Mark Idle.

b3 — Frame Complete/TDRA Select (FC/TDRA Select) — The FC/TDRA Select bit selects TDRA status or FC status for the TDRA/FC status bit indication. 1...FC status, 0...TDRA status.

b4 — Transmit Last Data (Tx Last) — Tx Last bit provides another method to terminate a frame. This bit should be set after loading the last data byte and before the Tx FIFO empties. When the Tx Last bit is set, the ADLC assumes the byte is the last byte and terminates the frame by appending CRCC and a closing Flag. This control bit is useful for DMA operation. Tx Last bit automatically returns to the "0" state.

b5 — Clear Receiver Status (CLR Rx ST) — When a "1" is written into the CLR Rx ST bit, a reset signal is generated for the receiver status bits in status registers #1 and #2 (except AP and RDA bits). The reset signal is enabled only for the bits which have been present during the last "read status" operation. The CLR Rx ST bit automatically returns to the "0" state.

b6 — Clear Transmitter Status (CLR Tx ST) — When a "1" is written into CLR Tx ST bit, a reset signal is generated for the transmitter status bits in status register #1 (except TDRA). The reset signal is enabled for the bits which have been present during the last "read status" operation. The CLR Tx ST bit automatically returns to the "0" state.

b7 — Request-to-Send Control (RTS) — The RTS bit, when high, causes the $\overline{RTS}$ output to be low (the active state). When the RTS bit returns low and data is being transmitted, the $\overline{RTS}$ output remains low until the last character of the frame (the closing Flag or Abort) has been completed and the Tx FIFO is empty. If the transmitter is idling when the RTS bit returns low, the $\overline{RTS}$ output will go high (the inactive state) within two bit times.

CONTROL REGISTER 3 (CR3)

				7	6	5	4	3	2	1	0
RS1	RS0	R/$\overline{W}$	AC	LOC/	GAP/	Loop	FDSE	01/11	AEX	CEX	LCF
0	1	0	1	DTR	TST			Idle			

b0 — Logical Control Field Select (LCF) — The LCF select bit causes the first byte(s) of data belonging to the information field to remain 8-bit characters until the logical control field is complete. The logical control field (when selected) is an automatically extendable field which is extended when bit 7 of a logical control character is a "1." When the LCF Select bit is reset the ADLC assumes no logical control field is present for either the transmit or received data channels. When the logical control field is terminated, the word length of the information data is then defined by WLS_1 and WLS_2.

b1 — Extended Control Field Select (C_{EX}) — When the C_{EX} bit is a "1", the control field is extended and asusmed to be 16 bits. When C_{EX} is "0", the control field is assumed to be 8 bits.

b2 — Auto/Address Extend Mode (A_{EX}) — The A_{EX} bit when "low" allows full 8 bits of the address octet to be utilized for addressing because address extension is inhibited. When the A_{EX} bit is "high," bit 0 of address octet equal to "0" causes the Address field to be extended by one octet. The exception to this automatic address field extension is when the first address octet is all "0's" (the Null Address).

b3 — 01/11 Idle (01/11 Idle) — The 01/11 Idle Control bit determines whether the inactive (Mark) idle condition begins with a "0" or not. If the 01/11 Idle Control is SET, the closing flag (or Abort) will be followed by a 011111...pattern. This is required of the controller for the "Go Ahead" character in the Loop Mode. When 01/11 is RESET, the idling condition will be all "1's".

b4 — Flag Detect Status Enable (FDSE) — The FDSE bit enables the FD status bit in Status Register #1 to indicate the occurrence of a received Flag character. The status indication will be accompanied by an interrupt if RIE is SET. Flag detection will cause the Flag Detect output to go low for 1 bit time regardless of the state of FDSE.

b5 — LOOP/NON-LOOP Mode (LOOP) — When the LOOP bit is set, loop mode operation is selected and the GAP/TST control bit, LOC/DTR control bit and LOC/DTR output are selected to perform the loop control functions. When LOOP is reset, the ADLC operates in the point-to-point data communications mode.

b6 — Go Active On Poll/Test (GAP/TST) — *In the Loop Mode* — The GAP/TST bit is used to respond to the poll sequence and to begin transmission. When GAP/TST is set, the receiver searches for the "Go Ahead" (or End of Poll, EOP). The receiver "Go ahead" is converted to an opening Flag and the ADLC starts its own transmission. When GAP/TST is reset during the transmission, the end of the frame (the completion of Flag or Abort) causes the termination of the "go-active-on-poll" operation and the Rx Data to Tx Data link is re-established. The ADLC then returns to the "loop-on-line" state.

In the Non-Loop Mode — The GAP/TST bit is used for self-test purposes. If GAP/TST bit is set, the TxD output is connected to the RxD input internally, and provides a "loop-back" feature. For normal operation, the GAP/TST bit should be reset.

b7 — Loop On-Line Control/DTR Control (LOC/DTR) — *In the Loop Mode* — The LOC/DTR bit is used to go on-line or to go off-line. When LOC/DTR is set, the ADLC goes to the on-line state after 7 consecutive "1's" occur at the RxD input. When LOC/DTR is reset, the ADLC goes to the "off-line" state after eight consecutive "1's" occur at the RxD input.

In the Non-Loop Mode — The LOC/DTR bit directly controls the Loop On-Line/$\overline{DTR}$ output state. 1...$\overline{DTR}$ output goes to low level, 0...$\overline{DTR}$ output goes to high level.

CONTROL REGISTER 4 (CR4)

				7	6	5	4	3	2	1	0
RS1	RS0	R/$\overline{W}$	AC	NRZI/NRZ	ABT_{EX}	ABT	Rx		Tx		"FF"/F
1	1	0	1				WLS_2	WLS_1	WLS_2	WLS_1	

b0 — Double Flag/Single Flag Interframe Control ("FF"/"F") — The "FF"/"F" Control bit determines whether the transmitter will transmit separate closing and opening Flags when frames are transmitted successively. When the "FF"/"F" control bit is low, the closing flag of the first frame will serve as the opening flag of the second frame. When the bit is high, independent opening and closing flags will be transmitted.

b1, b2 — Transmitter Word Length Select (Tx WLS1 and WLS2) — Tx WLS1 and WLS2 are used to select the word length of the transmitter information field. The encoding format is shown in Table 3.

b3, b4 — Receiver Word Length Select (Rx WLS1 and WLS2) — Rx WLS1 and WLS2 are used to select the word length of the receiver information field. The encoding format is shown in Table 3.

TABLE 3 — I-FIELD CHARACTER LENGTH SELECT

WLS_1	WLS_2	I-Field Character Length
0	0	5 bits
1	0	6 bits
0	1	7 bits
1	1	8 bits

b5 — Transmit Abort (ABT) — The ABT bit causes an Abort (at least 8 bits of "1" in succession) to be transmitted. The Abort is initiated and the Tx FIFO is cleared when the control bit goes high. Once Abort begins, the Tx Abort control bit assumes the low state.

b6 — Abort Extend (ABT_{EX}) — If ABT_{EX} is set, the abort code initiated by ABT is extended up to at least 16 bits of consecutive "1's", the mark Idle State.

b7 — NRZI (Zero Complement)/NRZ Select (NRZI/NRZ) — NRZI/NRZ bit selects the transmit/receive data format to be NRZI or NRZ in both Loop Mode or Non-Loop mode operation. When the NRZI Mode is selected, a 1-bit delay is added to the transmitted data (TxD) to allow for NRZI encoding. 1...NRZI, 0...NRZ.

NOTE

NRZI coding — The serial data remains in the same state to send a binary "1" and switches to the opposite state to send a binary "0".

STATUS REGISTER

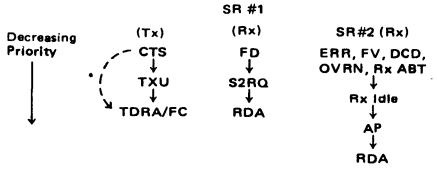
The Status Register #1 is the main status register. The IRQ bit indicates whether the ADLC requests service or not. The S2RQ bit indicates whether any bits in status register #2 request any service. TDRA and RDA, because they are most often used, are located in bit positions that are more convenient to test. RDA reflects the state of the RDA bit in status register #2.

The Status Register #2 provides the detailed status information contained in the S2RQ bit and these bits reflect receiver status. The FD bit is the only receiver status which is not indicated in status register #2.

The prioritized status mode provides maximum efficiency in searching the status bits and indicates only the most important action required to service the ADLC. The priority trees of both status registers are provided in Figure 10.

Reading the status register is a non-destructive process. The method of clearing status depends upon the bit's function and is discussed for each bit in the register.

FIGURE 10 — STATUS REGISTER PRIORITY TREE (PSE = 1)



*Prioritized even when PSE = 0
NOTE: Status bit above will inhibit one below it.

STATUS REGISTER 1 (SR1)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
				IRQ	TDRA/FC	TXU	CTS	FD	LOOP	S2RQ	RDA
0	0	1	X								

b0 — Receiver Data Available (RDA) — The RDA status bit reflects the state of the RDA status bit in status register #2. It provides the means of achieving data transfers of received data in the full Duplex Mode without having to read both status registers.

b1 — Status Register #2 Read Request (S2RQ) — All the status bits (stored conditions) of status register #2 (except RDA bit) are logically ORed and indicated by the S2RQ status bit. Therefore, S2RQ indicates that status register #2 needs to be read. When S2RQ is "0", it is not necessary to read status register #2. The bit is cleared when the appropriate bits in status register #2 are cleared or when Rx Reset is used.

b2 — Loop Status (LOOP) — The LOOP status bit is used to monitor the loop operation of the ADLC. This bit does not cause an IRQ. When Non-Loop Mode is selected, LOOP bit stays "0". When Loop Mode is selected, the LOOP status bit goes to "1" during "On-Loop" condition. When ADLC is in an "Off-Loop" condition or "Go-Active-On-Poll" condition, the LOOP status bit is a "0".

b3 — Flag Detected (FD) — The FD Status bit indicates that a flag has been received if the Flag Detect Enable control bit has been set. The bit goes high at the last bit of the Flag Character received (when the Flag Detect Output goes low) and is stored until cleared by Clear Rx Status or Rx Reset.

b4 — Clear-to-Send (CTS) — The $\overline{CTS}$ input positive transition is stored in the status register and causes an IRQ (if Enabled). The stored CTS condition and its IRQ are cleared by Clear Tx Status control bit or Tx Reset bit. After the stored status is reset, the CTS status bit reflects the state of the $\overline{CTS}$ input.

b5 — Transmitter Underrun (TxU) — When the transmitter runs out of data during a frame transmission, an underrun occurs and the frame is automatically terminated by transmitting an Abort. The underrun condition is indicated by the TxU status bit. TxU can be cleared by means of the Clear Tx Status Control bit or by Tx Reset.

b6 — Transmitter Data Register Available/Frame Complete (TDRA/FC) — The TDRA Status bit serves two purposes depending upon the state of the Frame Complete/TDRA Select control bit. When this bit serves as a TDRA status bit, it indicates that data (to be transmitted) can be loaded into the Tx Data FIFO register. The first register (Register #1) of the Tx Data FIFO being empty (TDRA = "1") will be indicated by the TDRA Status bit in the "1-Byte Transfer Mode." The first two registers (Registers #1 and #2) must be empty for TDRA to be high when in the "2-Byte Transfer Mode." TDRA is inhibited by Tx Reset, or $\overline{CTS}$ being high.

When the Frame Complete Mode of operation is selected, the TDRA/FC status bit goes high when an abort is transmitted or when a flag is transmitted with no data in the Tx FIFO. The bit remains high until cleared by resetting the TDRA/FC control bit or setting the Tx Reset bit.

b7 — Interrpt Request (IRQ) — The Interrupt Request status bit indicates when the $\overline{IRQ}$ output is in the active state ($\overline{IRQ}$ Output = "0"). The IRQ status bit is subject to the same interrupt enables (RIE, TIE) as the $\overline{IRQ}$ output, i.e., with both transmitter and receiver interrupts enabled, the IRQ status bit is a logical ORed indication of Status Register 1 status bits. The IRQ bit only reflects the set status bits which have interrupts enabled. The IRQ status bit simplifies status inquiries for polling systems by providing single bit indication of service requests.

STATUS REGISTER 2 (SR2)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
0	1	1	X	RDA	OVRN	DCD	ERR	Rx ABT	Rx Idle	FV	AP

b0 — Address Present (AP) — The AP status bit provides the frame boundary and indicates an Address octet is available in the Rx Data FIFO register. In the Extended Addressing Mode, the AP bit continues to indicate addresses until the Address field is complete. The Address present status bit is cleared by reading data or by Rx Reset.

b1 — Frame Valid (FV) — The FV status bit provides the frame boundary indication to the MPU and also indicates that a frame is complete with no error. The FV status bit is set when the last data byte of a frame is transferred into the last location of the Rx FIFO (available to be read by MPU). Once FV status is set, the ADLC stops further data transfer into the last location of the Rx FIFO (in order to prevent the mixing of two frames) until the status bit is cleared by the Clear Rx Status bit or Rx Reset.

b2 — Inactive Idle Received (Rx Idle) — The Rx Idle status bit indicates that a minimum of 15 consecutive "1's" have been received. The event is stored within the status register and can cause an interrupt. The interrupt and stored condition are cleared by the Clear Rx Status Control bit. The Status bit is the Logical OR of the receiver idling detector (which continues to reflect idling until a "0" is received) and the stored inactive idle condition.

b3 — Abort Received (RxABT) — The RxABT status bit indicates that seven or more consecutive "1's" have been received. Abort has no meaning under out-of-frame conditions; therefore, no interrupt nor storing of the status will occur unless a Flag has been detected prior to the Abort. An Abort Received when "in frame" is stored in the status register and causes an IRQ. The status bit is the logical OR of the stored conditions and the Rx Abort detect logic, which is cleared after 15 consecutive "1's" have occurred. The stored Abort condition is cleared by the Clear Rx Status Control bit or Rx Reset.

b4 — Frame Check Sequence/Invalid Frame Error (ERR) — When a frame is complete with a cyclic redundancy check (CRC) error or a short frame error (the frame does not have complete Address and Control fields), the ERR status bit is set instead of the Frame Valid status bit. Other functions, frame boundry indication and control function, are exactly the same as for the Frame Valid status bit. Refer to the FV status bit.

b5 — Data Carrier Detect (DCD) — A positive transition on the $\overline{DCD}$ input is stored in the status register and causes an IRQ (if enabled). The stored DCD condition and its IRQ are cleared by the Clear Rx Status Control bit or RX Reset. After stored status is reset, the DCD status bit follows the state of the input. Both the stored DCD condition and the $\overline{DCD}$ input cause the reset of the receiver section when they are high.

b6 — Receiver Overrun (OVRN) — OVRN status indicates that receiver data has been transferred into the Rx FIFO when it is full, resulting in data loss. The OVRN status is cleared by the Clear Rx Status bit or Rx Reset. Continued overrunning only destroys data in the first FIFO Register.

b7 — Receiver Data Available (RDA) — The Receiver Data Available status bit indicates when receiver data can be read from the Rx Data FIFO. When the prioritized status mode is used, the RDA bit indicates that non-address and non-last data are available in the Rx FIFO. The receiver data being present in the last register of the FIFO causes RDA to be high for the "1-Byte Transfer Mode." The RDA bit being high indicates that the last two registers are full when in the "2-Byte Transfer Mode." The RDA status bit is reset automatically when data is not available.

PROGRAMMING CONSIDERATIONS

1. **Status Priority** – When the prioritized status mode is used, it is best to test for the lowest priority conditions first. The lowest priority conditions typically occur more frequently and are the most likely conditions to exist when the processor is interrupted.
2. **Stored vs Present Status** – Certain status bits (DCD, CTS, Rx Abort, and Rx Idle) indicate a status which is the logical OR of a stored and a present condition. It is the stored status that causes an interrupt and which is cleared by a Status Clear control bit. After being cleared, the status register will reflect the present condition of an input or a receiver input sequence.
3. **Clearing Status Registers** – In order to clear an interrupt with the two Status Clear control bits, a particular status condition must be read before it can be cleared. In the prioritized mode, clearing a higher priority condition might result in another $\overline{IRQ}$ caused by a lower priority condition whose status was suppressed when a status register was first read. This guarantees that a status condition is never inadvertently cleared.
4. **Clearing the Rx FIFO** – An Rx Reset will effectively clear the contents of all three Rx FIFO bytes. However, the FIFO may contain data from two different frames when abort or DCD failure occurs. When this happens, the data from a previously closed frame (a frame whose closing flag has been received) will not be destroyed.
5. **Servicing the Rx FIFO in a 2-Byte Mode** – The procedure for reading the last bytes of data is the same, regardless of whether the frame contains an even or an odd number of bytes. Continue to read 2 bytes until an interrupt occurs that is caused by an end-of-frame status (FV or ERR). When this occurs, indicating the last byte either has been read or is ready to be read, switch temporarily to the 1-byte mode with no prioritized status (control register 2). Test RDA to indicate whether a 1-byte read should be performed. Then clear the frame end status.
6. **Frame Complete Status and $\overline{RTS}$ Release** – In many cases, a MODEM will require a delay for releasing $\overline{RTS}$. An 8-bit or 16-bit delay can be added to the ADLC $\overline{RTS}$ output by using an Abort. At the end of a transmission, frame complete status will indicate the frame completion. After frame complete status goes high, write "1" into the Abt control bit (and Abt Extend bit if a 16-bit delay is required). After the Abt control bit is set, write "0" into the $\overline{RTS}$ control bit. The transmitter will transmit eight or sixteen "1's" and the $\overline{RTS}$ output will then go high (inactive).
7. **Note to users not using the EF6800** – (a) Carte should be taken when performing a write followed by a read on successive E pulses at a high frequency rate. Time must be allowed for status changes to occur. If this is done, the time that E is low between successive write/read E pulses should be at least 500 ns. (b) The ADLC is a completely static part. However, the E frequency should be high enough to move data through the FIFOs and to service the peripheral requirements. Also, the period between successive E pulses should be less than the period of RxC or TxC in order to maintain synchronization between the data bus and the peripherals.
8. **Clear-to-Send ($\overline{CTS}$)** – The $\overline{CTS}$ input, when high, provides a real-time inhibit to the TDRA status bit and its associated interrupt. All other status bits will be operational. Since it inhibits TDRA, $\overline{CTS}$ also inhibts the TDSR DMA request. The $\overline{CTS}$ input being high does not affect any other part of the transmitter. Information in the Tx FIFO and Tx Shift Register will, therefore, continue to be transmitted as long as the Tx CLK is running.

ORDERING INFORMATION

EF68A54 | C | M | B/B

- EF68A54 — Device
- C — Package
- M — Oper. temp.
- B/B — Screening level

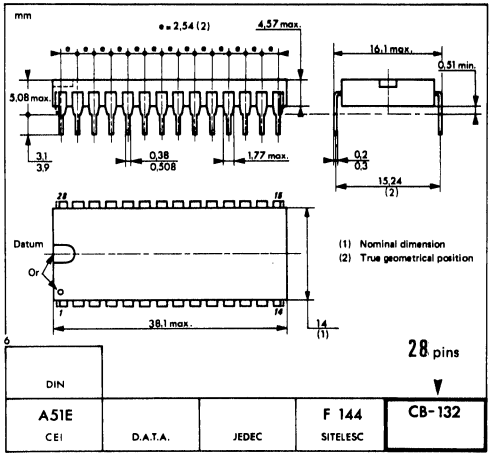
The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

DEVICE	PACKAGE					OPER. TEMP.			SCREENING LEVEL			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6854 (1.0 MHz)	●				●	●			●			
	●	●					●		●			
	●			●				●	●		●	●
		●						●	●		●	
EF68A54 (1.5 MHz)	●					●			●			
	●	●					●		●			
	●			●				●	●		●	●
			●					●	●		●	
EF68B54 (2.0 MHz)	●					●			●			
	●	●					●		●		●	

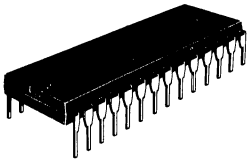
Examples : EF6854C, EF6854CV, EF6854CM, EF6854JM

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** −40°C to +85°C, **M :** −55°C to +125°C, ***** : may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.

PHYSICAL DIMENSIONS



CB-132



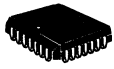
P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

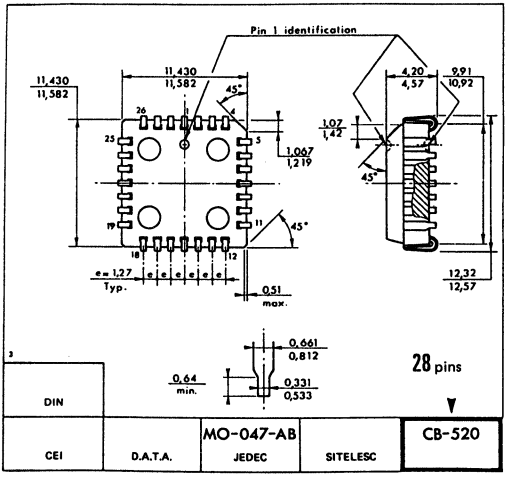
J SUFFIX
CERDIP PACKAGE

C SUFFIX
CERAMIC PACKAGE

CB-520

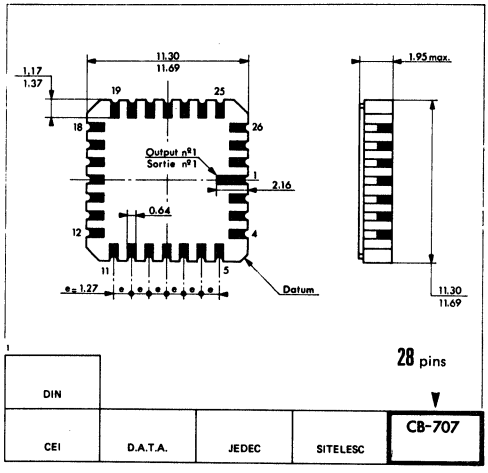


FN SUFFIX
PLCC 28



CB-707

E SUFFIX
LCCC 28



CHAPTER 3 - 68000 MICROPROCESSORS

68000 MICROPROCESSORS SELECTION GUIDE

Part number	Description	Technology	Alt source	CLK freq. (MHz)	Page
TS68000-8	16-bit MPU 32-bit data and address registers -	HMOS	MC68000-8	8	3-3
TS68000-10	16 megabyte direct addressing range -		MC68000-10	10	
TS68000-12	56 powerful instruction pages -		MC68000-12	12,5	
TS68000-16	Memory mapped I/O - 14 addressing modes			16	
TS68008-8	8-bit data bus version of TS68000	HMOS	MC68008-8	8	3-95
TS68008-10	1 Mbyte direct addressing space - Complete		MC68008-10	10	
TS68008-12	code compatibility with the TS68000		MC68008-12	12,5	
MK68201/04-4	16-bit MCU ROMLESS (UPC)	NMOS	—	4	3-191
MK68201/04-6	16-bit MCU ROMLESS (UPC)	"	"	6	
MK68211/04-4	16-bit MCU ROMLESS (GP)	"	"	4	
MK68211/04-6	16-bit MCU ROMLESS (GP)	"	"	6	
MK68E201/04-4	16-bit Emulator (UPC)	"	"	4	
MK68E211/04-4	16-bit Emulator (GP)	"	"	4	
MK68HC201/04-8	16-bit MCU ROMLESS (UPC)	HCMOS	"	8	3-265
MK68HC201/04-10	16-bit MCU ROMLESS (UPC)	"	"	10	
MK68HC201/04-12	16-bit MCU ROMLESS (UPC)	"	"	12.5	
MK68HC211/04-8	16-bit MCU ROMLESS (GP)	"	"	8	
MK68HC211/04-10	16-bit MCU ROMLESS (GP)	"	"	10	
MK68HC211/04-12	16-bit MCU ROMLESS (GP)	"	"	12.5	
MK68HCE221/08-8	16-bit Emulator (UPC/GP)	"	"	8	
MK68HCE221/08-10	16-bit Emulator (UPC/GP)	"	"	10	
MK68HCE221/08-12	16-bit Emulator (UPC/GP)	"	"	12.5	

THOMSON COMPONENTS MOSTEK

TS68000

16-/32-BIT MICROPROCESSOR

ADVANCE INFORMATION

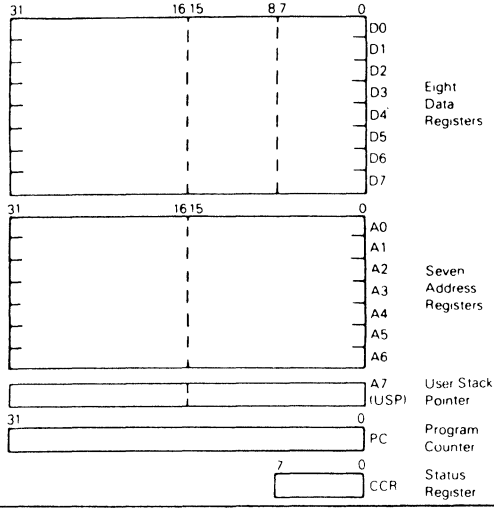
The TS68000 is the first implementation of the 68000 16/32 microprocessor architecture. The TS68000 has a 16-bit data bus and 24-bit address bus while the full architecture provides for 32-bit address and data buses. It is completely code-compatible with the TS68008 8-bit data bus implementation of the 68000 and is downward code-compatible with the TS68020 32-bit implementation of the architecture. Any user-mode programs written using the TS68000 instruction set will run unchanged on the TS68008 and TS68020. This is possible because the user programming model is identical for all three processors and the instruction sets are proper sub-sets of the complete architecture.

The resources available to the TS68000 user consist of the following :

- 16 32-bit data and address registers
- 16 megabyte direct addressing range
- 56 powerful instruction types
- Operations on five main data types
- Memory mapped I/O
- 14 addressing modes
- 4 available versions : 8 MHz, 10 MHz, 12.5 MHz and 16 MHz.

As shown in the user programming model, the TS68000 offers 16 32-bit registers and a 32-bit program counter. The first eight registers (D0-D7) are used as data registers for byte (8-bit), word (16-bit), and long word (32-bit) operations. The second set of seven registers (A0-A6) and the user stack pointer (USP) may be used as software stack pointers and base address registers. In addition, the registers may be used for word and long word operations. All of the 16 registers may be used as index registers.

PROGRAMMING MODEL

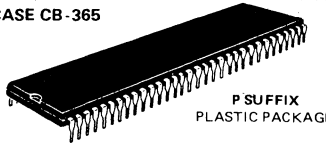


HMOS

HIGH-DENSITY, N-CHANNEL, SILICON-GATE, DEPLETION LOAD

16-/32-BIT MICROPROCESSOR

CASE CB-365

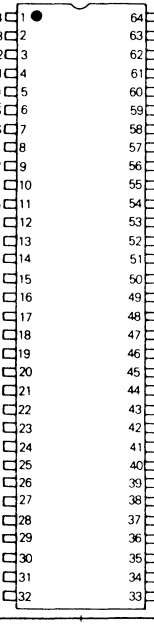


P SUFFIX
PLASTIC PACKAGE

CASE CB-523

FN SUFFIX
PLCC 68

PIN ASSIGNMENT



SECTION 1
INTRODUCTION

The TS68000 is the first implementation of the 68000 16/32 microprocessor architecture. The TS68000 has a 16-bit data bus and 24-bit address bus while the full architecture provides for 32-bit address an data buses. It is completely code-compatible with the TS68008 8-bit data bus implementation of the 68000 and is downward code-compatible with the TS68020 32-bit implementation of the architecture. Any user-mode programs written using the TS68000 instruction set will run unchanged on the TS68008 and TS68020. This is possible because the user programming model is identical for all four processors and the instruction sets are proper sub-sets of the complete architecture.

The resources available to the TS68000 user consist of the following :

- 17 32-Bit Data and Address Registers
- 16 Megabyte Direct Addressing Range
- 56 Powerful Instruction Types
- Operations on Five Main Data Types
- Memory Mapped I/O
- 14 Addressing Modes

As shown in the user programming model (Figure 1-1), the TS68000 offers 16 32-bit registers and a 32-bit program counter. The first eight registers (D0-D7) are used as data registers for byte (8-bit), word (16-bit), and long word (32-bit) operations. The second set of seven registers (A0-A6) and the user stack pointer (USP) may be used as software stack pointers and base address registers. In addition, the registers may be used for word and long word operations. All of the 16 registers may be used as index registers.

In supervisor mode, the upper byte of the status register and the supervisor stack pointer (SSP) are also available to the programmer. These registers are shown in Figure 1-2.

The status register (Figure 1-3) contains the interrupt mask (eight levels available) as well as the condition codes: extend (X), negative (N), zero (Z), overflow (V), and carry (C). Additional status bits indicate that the processor is in a trace (T) mode and in a supervisor (S) or user state.

1.1 DATA TYPES AND ADDRESSING MODES

Five basic data types are supported. These data types are:

- Bits
- BCD Digits (4 bits)
- Bytes (8 bits)
- Words (16 bits)
- Long Words (32 bits)

In addition, operations on other data types such as memory addresses, status word data, etc., are provided in the instruction set.

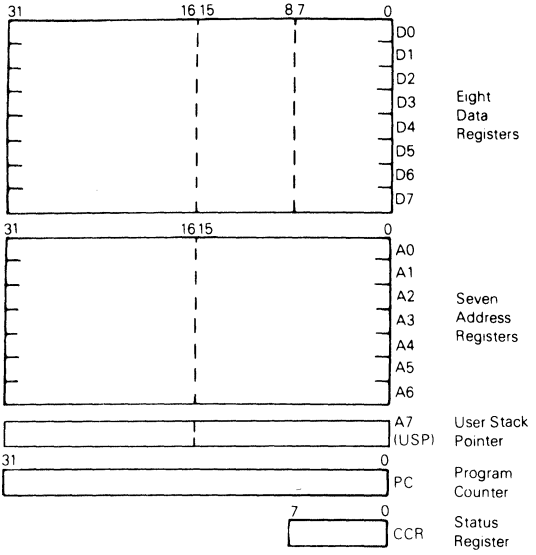


Figure 1-1. User Programming Model

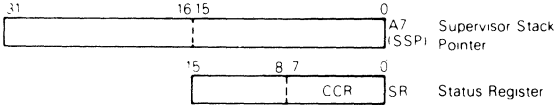


Figure 1-2. Supervisor Programming Model Supplement

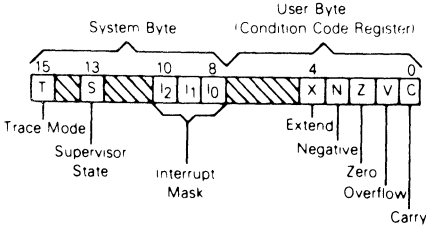


Figure 1-3. Status Register

The 14 address modes, shown in Table 1-1, include six basic types:

- Register Direct
- Register Indirect
- Absolute
- Program Counter Relative
- Immediate
- Implied

Included in the register indirect addressing modes is the capability to do postincrementing, predecrementing, offsetting, and indexing. The program counter relative mode can also be modified via indexing and offsetting.

Table 1-1. Addressing Modes

Addressing Modes	Syntax
Register Direct Addressing	
Data Register Direct	Dn
Address Register Direct	An
Absolute Data Addressing	
Absolute Short	xxx.W
Absolute Long	xxx.L
Program Counter Relative Addressing	
Relative with Offset	d_{16}(PC)
Relative with Index Offset	d_8(PC,Xn)
Register Indirect Addressing	
Register Indirect	(An)
Postincrement Register Indirect	(An)+
Predecrement Register Indirect	−(An)
Register Indirect with Offset	d_{16}(An)
Indexed Register Indirect with Offset	d_8(An,Xn)
Immediate Data Addressing	
Immediate	#xxx
Quick Immediate	#1-#8
Implied Addressing	
Implied Register	SR USP SP PC

NOTES:
Dn = Data Register
An = Address Register
Xn = Address or Data Register used as Index Register
SR = Status Register
PC = Program Counter
SP = Stack Pointer
USP = User Stack Pointer
( ) = Effective Address
d_8 = 8-Bit Offset (Displacement)
d_{16} = 16-Bit Offset (Displacement)
#xxx = Immediate Data

1.2 INSTRUCTION SET OVERVIEW

The TS68000 instruction set is shown in Table 1-2. Some additional instructions are variations, or subsets, of these and they appear in Table 1-3. Special emphasis has been given to the instruction set's support of structured high-level languages to facilitate ease of programming. Each instruction, with few exceptions, operates on bytes, words, and long words and most instructions can use any

of the 14 addressing modes. Combining instruction types, data types, and addressing modes, over 1000 useful instructions are provided. These instructions include signed and unsigned, multiply and divide, "quick" arithmetic operations, BCD arithmetic, and expanded operations (through traps).

Table 1-2. Instruction Set Summary

Mnemonic	Description
ABCD	Add Decimal With Extend
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
Bcc	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DBcc	Test Condition, Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive Or
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical Or
PEA	Push Effective Address
RESET	Reset External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

Table 1-3. Variations of Instruction Types

Instruction Type	Variation	Description
ADD	**ADD**	Add
	ADDA	Add Address
	ADDQ	Add Quick
	ADDI	Add Immediate
	ADDX	Add with Extend
AND	**AND**	Logical And
	ANDI	And Immediate
	ANDI to CCR	And Immediate to Condition Codes
	ANDI to SR	And Immediate to Status Register
CMP	**CMP**	Compare
	CMPA	Compare Address
	CMPM	Compare Memory
	CMPI	Compare Immediate
EOR	**EOR**	Exclusive Or
	EORI	Exclusive Or Immediate
	EORI to CCR	Exclusive Or Immediate to Condition Codes
	EORI to SR	Exclusive Or Immediate to Status Register

Instruction Type	Variation	Description
MOVE	**MOVE**	Move
	MOVEA	Move Address
	MOVEM	Move Multiple Registers
	MOVEP	Move Peripheral Data
	MOVEQ	Move Quick
	MOVE from SR	Move from Status Register
	MOVE to SR	Move to Status Register
	MOVE to CCR	Move to Condition Codes
	MOVE USP	Move User Stack Pointer
NEG	**NEG**	Negate
	NEGX	Negate with Extend
OR	**OR**	Logical Or
	ORI	Or Immediate
	ORI to CCR	Or Immediate to Condition Codes
	ORI to SR	Or Immediate to Status Register
SUB	**SUB**	Subtract
	SUBA	Subtract Address
	SUBI	Subtract Immediate
	SUBQ	Subtract Quick
	SUBX	Subtract with Extend

SECTION 2
DATA ORGANIZATION AND ADDRESSING CAPABILITIES

This section contains a description of the registers and the data organization of the TS68000.

2.1 OPERAND SIZE

Operand sizes are defined as follows: a byte equals 8 bits, a word equals 16 bits, and a long word equals 32 bits. The operand size for each instruction is either explicitly encoded in the instruction or implicitly defined by the instruction operation. Implicit instructions support some subset of all three sizes.

2.2 DATA ORGANIZATION IN REGISTERS

The eight data registers support data operands of 1, 8, 16, or 32 bits. The seven address registers together with the stack pointers support address operands of 32 bits.

2.2.1 Data Registers

Each data register is 32 bits wide. Byte operands occupy the low order 8 bits, word operands the low order 16 bits, and long word operands the entire 32 bits. The least significant bit is addressed as bit zero; the most significant bit is addressed as bit 31.

When a data register is used as either a source or destination operand, only the appropriate low order portion is changed; the remaining high order portion is neither used nor changed.

2.2.2 Address Registers

Each address register and the stack pointer is 32 bits wide and holds a full 32-bit address. Address registers do not support the sized operands. Therefore, when an address register is used as a source operand, either the low order word or the entire long word operand is used depending upon the operation size. When an address register is used as the destination operand, the entire register is affected regardless of the operation size. If the operation size is word, any other operands are sign extended to 32 bits before the operation is performed.

2.3 DATA ORGANIZATION IN MEMORY

Bytes are individually addressable with the high order byte having an even address the same as the word, as shown in Figure 2-1. The low order byte has an odd address that is one count higher than the word address. Instructions and multibyte data are accessed only on word (even byte) boundaries. If a long word datum is located at address n (n even), then the second word of that datum is located at address n + 2.

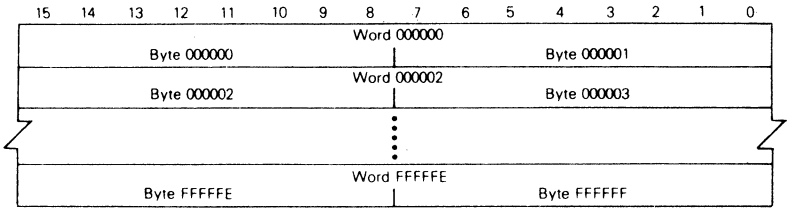


Figure 2-1. Word Organization in Memory

The data types supported by the TS68000 are : bit data, integer data of 8, 16, or 32 bits, 32-bit addresses and binary coded decimal data. Each of these data types is put in memory, as shown in Figure 2-2. The numbers indicate the order in which the data would be accessed from the processor.

2.4 ADDRESSING

Instructions for the TS68000 contain two kinds of information : the type of function to be performed and the location of the operand(s) on which to perform that function. The methods used to locate (address) the operand(s) are explained in the following paragraphs.

Instructions specify an operand location in one of three ways:

Register Specification – the number of the register is given in the register field of their instruction.

Effective Address – use of the different effective addressing modes.

Implicit Reference – the definition of certain instructions implies the use of specific registers.

2.5 INSTRUCTION FORMAT

Instructions are from one to five words in length as shown in Figure 2-3. The length of the instruction and the operation to be performed is specified by the first word of the instruction which is called the operation word. The remaining words further specify the operands. These words are either immediate operands or extensions to the effective address mode specified in the operation word.

2.6 PROGRAM/DATA REFERENCES

The TS68000 separates memory references into two classes : program references and data references. Program references, as the name implies, are references to that section of memory that contains the program being executed. Data references refer to that section of memory that contains data. Operand reads are from the data space except in the case of the program counter relative addressing mode. All operand writes are to the data space.

2.7 REGISTER SPECIFICATION

The register field within an instruction specifies the register to be used. Other fields within the instruction specify whether the register selected is an address or data register and how the register is to be used.

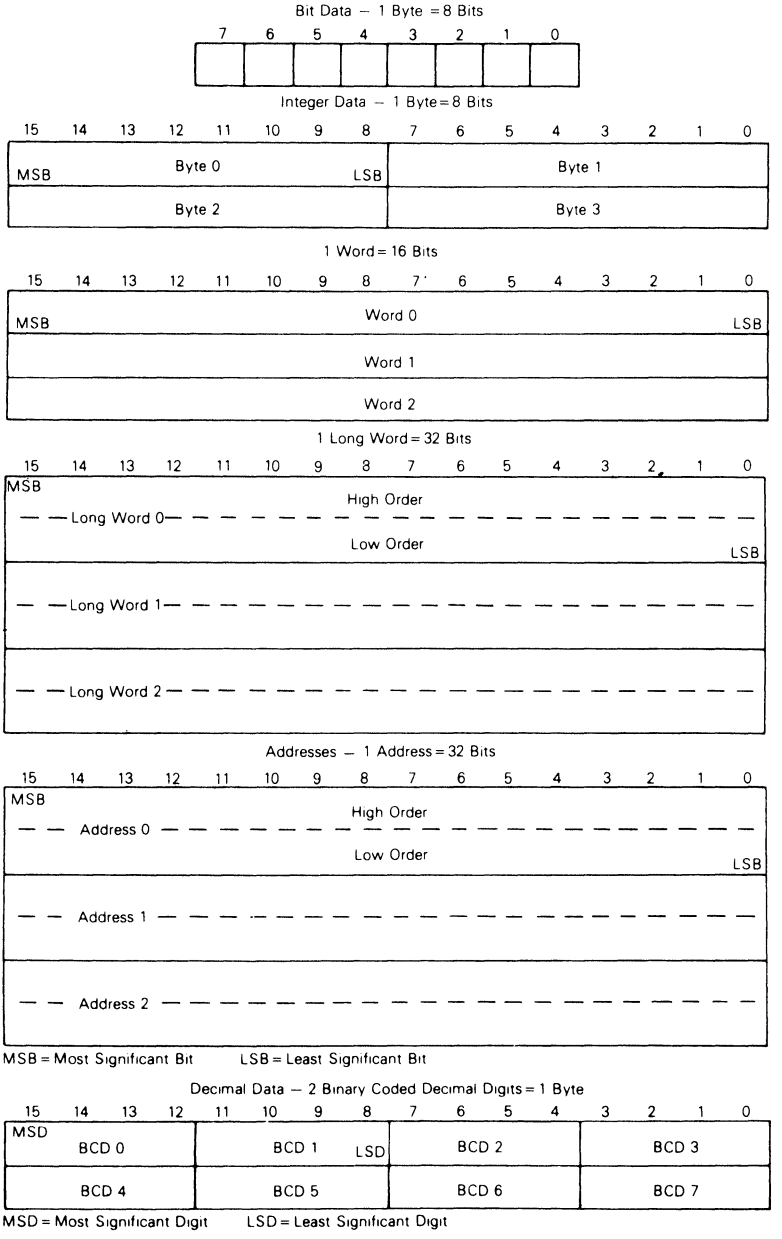


Figure 2-2. Memory Data Organization

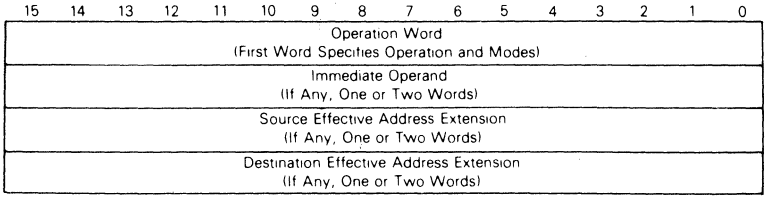


Figure 2-3. Instruction Operation Word General Format

2.8 EFFECTIVE ADDRESS

Most instructions specify the location of an operand by using the effective address field in the operation word. For example, Figure 2-4 shows the general format of the single-effective-address instruction operation word. The effective address is composed of two 3-bit fields: the mode field and the register field. The value in the mode field selects the different address modes. The register field contains the number of a register.

The effective address field may require additional information to fully specify the operand. This additional information, called the effective address extension, is contained in the following word or words and is considered part of the instruction, as shown in Figure 2-3. The effective address modes are grouped into three categories: register direct, memory addressing, and special.

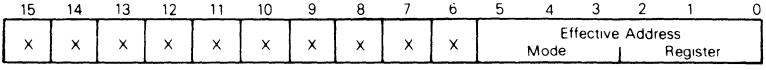


Figure 2-4. Single-Effective-Address Instruction Operation Word

2.8.1 Register Direct Modes

These effective addressing modes specify that the operand is in one of 16 multifunction registers.

2.8.1.1 DATA REGISTER DIRECT. The operand is in the data register specified by the effective address register field.

2.8.1.2 ADDRESS REGISTER DIRECT. The operand is in the address register specified by the effective address register field.

2.8.2 Memory Address Modes

These effective addressing modes specify that the operand is in memory and provide the specific address of the operand.

2.8.2.1 ADDRESS REGISTER INDIRECT. The address of the operand is in the address register specified by the register field. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.2.2 ADDRESS REGISTER INDIRECT WITH POSTINCREMENT. The address of the operand is in the address register specified by the register field. After the operand address is used, it is incremented by one, two, or four depending upon whether the size of the operand is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is incremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

2.8.2.3 ADDRESS REGISTER INDIRECT WITH PREDECREMENT. The address of the operand is in the address register specified by the register field. Before the operand address is used, it is decremented by one, two, or four depending upon whether the operand size is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is decremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

2.8.2.4 ADDRESS REGISTER INDIRECT WITH DISPLACEMENT. This addressing mode requires one word of extension. The address of the operand is the sum of the address in the address register and the sign-extended 16-bit displacement integer in the extension word. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.2.5 ADDRESS REGISTER INDIRECT WITH INDEX. This addressing mode requires one word of extension. The address of the operand is the sum of the address in the address register, the sign-extended displacement integer in the low order eight bits of the extension word, and the contents of the index register. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.3 Special Address Modes

The special address modes use the effective address register field to specify the special addressing mode instead of a register number.

2.8.3.1 ABSOLUTE SHORT ADDRESS. This addressing mode requires one word of extension. The address of the operand is the extension word. The 16-bit address is sign extended before it is used. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.3.2 ABSOLUTE LONG ADDRESS. This addressing mode requires two words of extension. The address of the operand is developed by the concatenation of the extension words. The high order part of the address is the first extension word; the low order part of the address is the second extension word. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.3.3 PROGRAM COUNTER WITH DISPLACEMENT. This addressing mode requires one word of extension. The address of the operand is the sum of the address in the program counter and the sign-extended 16-bit displacement integer in the extension word. The value in the program counter is the address of the extension word. The reference is classified as a program reference.

2.8.3.4 PROGRAM COUNTER WITH INDEX. This addressing mode requires one word of extension. The address is the sum of the address in the program counter, the sign-extended displacement integer in the lower eight bits of the extension word, and the contents of the index register. The value in the program counter is the address of the extension word. This reference is classified as a program reference.

2.8.3.5 IMMEDIATE DATA. This addressing mode requires either one or two words of extension depending on the size of the operation.

Byte Operation – operand is low order byte of extension word

Word Operation – operand is extension word

Long Word Operation – operand is in the two extension words, high order 16 bits are in the first extension word, low order 16 bits are in the second extension word.

2.8.3.6 IMPLICIT REFERENCE. Some instructions make implicit reference to the program counter (PC), the system stack pointer (SP), the supervisor stack pointer (SSP), the user stack pointer (USP), or the status register (SR). A selected set of instructions may reference the status register by means of the effective address field. These are:

ANDI to CCR	EORI to SR	MOVE to CCR
ANDI to SR	ORI to CCR	MOVE to SR
EORI to CCR	ORI to SR	MOVE from SR

2.9 EFFECTIVE ADDRESS ENCODING SUMMARY

Table 2-1 is a summary of the effective addressing modes discussed in the previous paragraphs.

Table 2-1. Effective Address Encoding Summary

Addressing Mode	Mode	Register
Data Register Direct	000	Register Number
Address Register Direct	001	Register Number
Address Register Indirect	010	Register Number
Address Register Indirect with Postincrement	011	Register Number
Address Register Indirect with Predecrement	100	Register Number
Address Register Indirect with Displacement	101	Register Number

Addressing Mode	Mode	Register
Address Register Indirect with Index	110	Register Number
Absolute Short	111	000
Absolute Long	111	001
Program Counter with Displacement	111	010
Program Counter with Index	111	011
Immediate	111	100

2.10 SYSTEM STACK

The system stack is used implicitly by many instructions; user stacks and queues may be created and maintained through the addressing modes. Address register seven (A7) is the system stack pointer (SP). The system stack pointer is either the supervisor stack pointer (SSP) or the user stack pointer (USP), depending on the state of the S bit in the status register. If the S bit indicates supervisor state, SSP is the active system stack pointer and the USP cannot be referenced as an address register. If the S bit indicates user state, the USP is the active system stack pointer, and the SSP cannot be referenced. Each system stack fills from high memory to low memory.

SECTION 3
INSTRUCTION SET SUMMARY

This section contains an overview of the form and structure of the TS68000 instruction set. The instructions form a set of tools that include all the machine functions to perform the following operations:

Data Movement	Bit Manipulation
Integer Arithmetic	Binary Coded Decimal
Logical	Program Control
Shift and Rotate	System Control

The complete range of instruction capabilities combined with the flexible addressing modes described previously provide a very flexible base for program development.

3.1 DATA MOVEMENT OPERATIONS

The basic method of data acquisition (transfer and storage) is provided by the move (MOVE) instruction. The move instruction and the effective addressing modes allow both address and data manipulation. Data move instructions allow byte, word, and long word operands to be transferred from memory to memory, memory to register, register to memory, and register to register. Address move instructions allow word and long word operand transfers and ensure that only legal address manipulations are executed. In addition to the general move instruction there are several special data movement instructions: move multiple registers (MOVEM), move peripheral data (MOVEP), exchange registers (EXG), load effective address (LEA), push effective address (PEA), link stack (LINK), unlink stack (UNLK), and move quick (MOVEQ). Table 3-1 is a summary of the data movement operations.

Table 3-1. Data Movement Operations

Instruction	Operand Size	Operation
EXG	32	Rx ↔ Ry
LEA	32	EA ← An
LINK	–	An → – (SP) SP → An SP + displacement → SP
MOVE	8, 16, 32	s → d
MOVEM	16, 32	(EA) → An, Dn An, Dn → EA
MOVEP	16, 32	(EA) → Dn Dn → (EA)
MOVEQ	8	#xxx → Dn
PEA	32	EA → – (SP)
SWAP	32	Dn[31:16] ↔ Dn[15:0]
UNLK	–	An → Sp (SP) + → An

NOTES:
s = source
d = destination
[ ] = bit number
– ( ) = indirect with predecrement
( ) + = indirect with postdecrement
= immediate data

3.2 INTEGER ARITHMETIC OPERATIONS

The arithmetic operations include the four basic operations of add (ADD), subtract (SUB), multiply (MUL), and divide (DIV) as well as arithmetic compare (CMP), clear (CLR), and negate (NEG). The add and subtract instructions are available for both address and data operations, with data operations accepting all operand sizes. Address operations are limited to legal address size operands (16 or 32 bits). Data, address, and memory compare operations are also available. The clear and negate instructions may be used on all sizes of data operands.

The multiply and divide operations are available for signed and unsigned operands using word multiply to produce a long word product, and a long word dividend with word divisor to produce a word quotient with a word remainder.

Multiprecision and mixed size arithmetic can be accomplished using a set of extended instructions. These instructions are: add extended (ADDX), subtract extended (SUBX), sign extend (EXT), and negate binary with extend (NEGX).

A test operand (TST) instruction that will set the condition codes as a result of a compare of the operand with zero is also available. Test and set (TAS) is a synchronization instruction useful in multiprocessor systems. Table 3-2 is a summary of the integer arithmetic operations.

Table 3-2. Integer Arithmetic Operations

Instruction	Operand Size	Operation
ADD	8, 16, 32	Dn + (EA) → Dn
		(EA) + Dn → (EA)
		(EA) + #xxx → (EA)
	16, 32	An + (EA) → An
ADDX	8, 16, 32	Dx + Dy + X → Dx
	16, 32	−(Ax) + −(Ay) + X → (Ax)
CLR	8, 16, 32	0 → EA
CMP	8, 16, 32	Dn − (EA)
		(EA) − #xxx
		(Ax) + − (Ay) −
	16, 32	An − (EA)
DIVS	32 ÷ 16	Dn ÷ (EA) → Dn
DIVU	32 ÷ 16	Dn ÷ (EA) → Dn
EXT	8 → 16	$(Dn)_8 \rightarrow Dn_{16}$
	16 → 32	$(Dn)_{16} \rightarrow Dn_{32}$
MULS	16 × 16 → 32	Dn × (EA) → Dn
MULU	16 × 16 → 32	Dn × (EA) → Dn
NEG	8, 16, 32	0 − (EA) → (EA)
NEGX	8, 16, 32	0 − (EA) − X → (EA)
SUB	8, 16, 32	Dn − (EA) → Dn
		(EA) − Dn → (EA)
		(EA) − #xxx → (EA)
	16, 32	An − (EA) → An
SUBX	8, 16, 32	Dx − Dy − X → Dx
		−(Ax) − −(Ay) − X → (Ax)
TAS	8	[EA] − 0, 1 → EA[7]
TST	8, 16, 32	(EA) − 0

NOTES:
[ ] = bit number
−( ) = indirect with predecrement
( )+ = indirect with postdecrement
= immediate data

3.3 LOGICAL OPERATIONS

Logical operation instructions AND, OR, EOR, and NOT are available for all sizes of integer data operands. A similar set of immediate instructions (ANDI, ORI, and EORI) provide these logical operations with all sizes of immediate data. Table 3-3 is a summary of the logical operations.

Table 3-3. Logical Operations

Instruction	Operand Size	Operation
AND	8, 16, 32	DnΛ(EA) → Dn (EA)ΛDn → (EA) (EA)Λ#xxx → (EA)
OR	8, 16, 32	Dn v (EA) → Dn (EA) v Dn → (EA) (EA) v #xxx → (EA)
EOR	8, 16, 32	(EA) ⊕ Dy → (EA) (EA) ⊕ #xxx → (EA)
NOT	8, 16, 32	~ (EA) → (EA)

NOTES:
~ = invert
# = immediate data
Λ = logical AND
V = logical OR
⊕ = logical exclusive OR

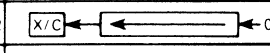
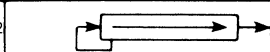
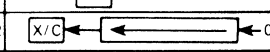
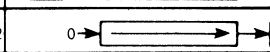
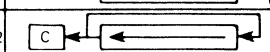
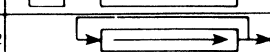
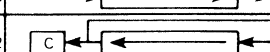
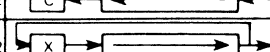
3.4 SHIFT AND ROTATE OPERATIONS

Shift operations in both directions are provided by the arithmetic instructions ASR and ASL and logical shift instructions LSR and LSL. The rotate instructions (with and without extend) available are ROXR, ROXL, ROR, and ROL. All shift and rotate operations can be performed in either registers or memory. Register shifts and rotates support all operand sizes and allow a shift count specified in a data register.

Memory shifts and rotates are for word operands only and allow only single-bit shifts or rotates.

Table 3-4 is a summary of the shift and rotate operations.

Table 3-4. Shift and Rotate Operations

Instruction	Operand Size	Operation
ASL	8, 16, 32	
ASR	8, 16, 32	
LSL	8, 16, 32	
LSR	8, 16, 32	
ROL	8, 16, 32	
ROR	8, 16, 32	
ROXL	8, 16, 32	
ROXR	8, 16, 32	

3.5 BIT MANIPULATION OPERATIONS

Bit manipulation operations are accomplished using the following instructions: bit test (BTST), bit test and set (BSET), bit test and clear (BCLR), and bit test and change (BCHG). Table 3-5 is a summary of the bit manipulation operations. (Z is bit 2 of the status register.)

Table 3-5. Bit Manipulation Operations

Instruction	Operand Size	Operation
BTST	8, 32	~ bit of (EA) → Z
BSET	8, 32	~ bit of (EA) → Z 1 → bit of EA
BCLR	8, 32	~ bit of (EA) → Z 0 → bit of EA
BCHG	8, 32	~ bit of (EA) → Z ~ bit of (EA) → bit of EA

NOTE: ~ = invert

3.6 BINARY CODED DECIMAL OPERATIONS

Multiprecision arithmetic operations on binary coded decimal numbers are accomplished using the following instructions: add decimal with extend (ABCD), subtract decimal with extend (SBCD), and negate decimal with extend (NBCD). Table 3-6 is a summary of the binary coded decimal operations.

Table 3-6. Binary Coded Decimal Operations

Instruction	Operand Size	Operation
ABCD	8	$Dx_{10} + Dy_{10} + X \rightarrow Dx$ $-(Ax)_{10} + -(Ay)_{10} + x \rightarrow (Ax)$
SBCD	8	$Dx_{10} - Dy_{10} - X \rightarrow Dx$ $-(Ax)_{10} - -(Ay)_{10} - X \rightarrow (Ax)$
NBCD	8	$0 - (EA)_{10} - X \rightarrow (EA)$

NOTE: −( ) = indirect with predecrement

3.7 PROGRAM CONTROL OPERATIONS

Program control operations are accomplished using a series of conditional and unconditional branch instructions and return instructions. These instructions are summarized in Table 3-7.

The conditional instructions provide setting and branching for the following conditions:

CC – carry clear
CS – carry set
EQ – equal
F – never true
GE – greater or equal
GT – greater than
HI – high
LE – less or equal
LS – low or same
LT – less than
MI – minus
NE – not equal
PL – plus
T – always true
VC – no overflow
VS – overflow

Table 3-7. Program Control Operations

Instruction	Operation
Conditional	
B_{CC}	Branch Conditionally (14 Conditions) 8- and 16-Bit Displacement
DB_{CC}	Test Condition, Decrement, and Branch 16-Bit Displacement
S_{CC}	Set Byte Conditionally (16 Conditions)
Unconditional	
BRA	Branch Always 8- and 16-Bit Displacement
BSR	Branch to Subroutine 8- and 16-Bit Displacement
JMP	Jump
JSR	Jump to Subroutine
Returns	
RTR	Return and Restore Condition Codes
RTS	Return from Subroutine

3.8 SYSTEM CONTROL OPERATIONS

System control operations are accomplished by using privileged instructions, trap generating instructions, and instructions that use or modify the status register. These instructions are summarized in Table 3-8.

Table 3-8. System Control Operations

Instruction	Operation
Privileged	
ANDI to SR	Logical AND to Status Register
EORI to SR	Logical EOR to Status Register
MOVE EA to SR	Load New Status Register
MOVE USP	Move User Stack Pointer
ORI to SR	Logical OR to Status Register
RESET	Reset External Devices
RTE	Return from Exception
STOP	Stop Program Execution
Trap Generating	
CHK	Check Data Register Against Upper Bounds
TRAP	Trap
TRAPV	Trap on Overflow
Status Register	
ANDI to CCR	Logical AND to Condition Codes
EORI to CCR	Logical EOR to Condition Codes
MOVE EA to CCR	Load New Condition Codes
MOVE SR to EA	Store Status Register
ORI to CCR	Logical OR to Condition Codes

SECTION 4
SIGNAL AND BUS OPERATION DESCRIPTION

This section contains a brief description of the input and output signals. A discussion of bus operation during the various machine cycles and operations is also given.

NOTE

The terms **assertion** and **negation** will be used extensively. This is done to avoid confusion when dealing with a mixture of "active-low" and "active-high" signals. The term assert or assertion is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. The term negate or negation is used to indicate that a signal is inactive or false.

4.1 SIGNAL DESCRIPTION

The input and output signals can be functionally organized into the groups shown in Figure 4-1. The following paragraphs provide a brief description of the signals and a reference (if applicable) to other paragraphs that contain more detail about the function being performed.

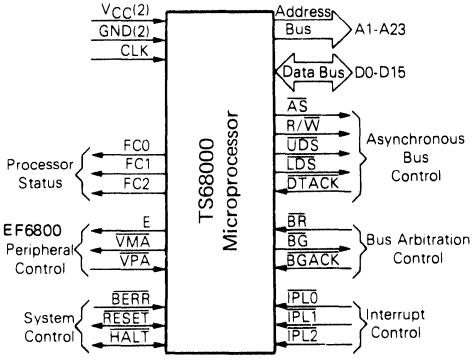


Figure 4-1. Input and Output Signals

4.1.1 Address Bus (A1 through A23)

This 23-bit, unidirectional, three-state bus is capable of addressing 8 megawords of data. It provides the address for bus operation during all cycles except interrupt cycles. During interrupt cycles, address lines A1, A2, and A3 provide information about what level interrupt is being serviced while address lines A4 through A23 are all set to a logic high.

4.1.2 Data Bus (D0 through D15)

This 16-bit, bidirectional, three-state bus is the general purpose data path. It can transfer and accept data in either word or byte length. During an interrupt acknowledge cycle, the external device supplies the vector number on data lines D0-D7.

4.1.3 Asynchronous Bus Control

Asynchronous data transfers are handled using the following control signals: address strobe, read/write, upper and lower data strobes, and data transfer acknowledge. These signals are explained in the following paragraphs.

4.1.3.1 ADDRESS STROBE ($\overline{AS}$). This signal indicates that there is a valid address on the address bus.

4.1.3.2 READ/WRITE (R/$\overline{W}$). This signal defines the data bus transfer as a read or write cycle. The R/$\overline{W}$ signal also works in conjunction with the data strobes as explained in the following paragraph.

4.1.3.3 UPPER AND LOWER DATA STROBE ($\overline{UDS}$, $\overline{LDS}$). These signals control the flow of data on the data bus, as shown in Table 4-1. When the R/$\overline{W}$ line is high, the processor will read from the data bus as indicated. When the R/$\overline{W}$ line is low, the processor will write to the data bus as shown.

Table 4-1. Data Strobe Control of Data Bus

$\overline{UDS}$	$\overline{LDS}$	R/$\overline{W}$	D8-D15	D0-D7
High	High	–	No Valid Data	No Valid Data
Low	Low	High	Valid Data Bits 8-15	Valid Data Bits 0-7
High	Low	High	No Valid Data	Valid Data Bits 0-7
Low	High	High	Valid Data Bits 8-15	No Valid Data
Low	Low	Low	Valid Data Bits 8-15	Valid Data Bits 0-7
High	Low	Low	Valid Data Bits 0-7*	Valid Data Bits 0-7
Low	High	Low	Valid Data Bits 8-15	Valid Data Bits 8-15*

* These conditions are a result of current implementation and may not appear on future devices.

4.1.3.4 DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$). This input indicates that the data transfer is completed. When the processor recognizes $\overline{DTACK}$ during a read cycle, data is latched and the bus cycle terminated. When $\overline{DTACK}$ is recognized during a write cycle, the bus cycle is terminated. (Refer to **4.4 ASYNCHRONOUR VERSUS SYNCHRONOUS OPERATION**).

4.1.4 Bus Arbitration Control

The three signals, bus request, bus grant, and bus grant acknowledge, form a bus arbitration circuit to determine which device will be the bus master device.

4.1.4.1 BUS REQUEST ($\overline{BR}$). This input is wire ORed with all other devices that could be bus masters. This input indicates to the processor that some other device desires to become the bus master.

4.1.4.2 BUS GRANT ($\overline{BG}$). This output indicates to all other potential bus master devices that the processor will release bus control at the end of the current bus cycle.

4.1.4.3 BUS GRANT ACKNOWLEDGE ($\overline{BGACK}$). This input indicates that some other device has become the bus master. This signal should not be asserted until the following four conditions are met:

1. a bus grant has been received,
2. address strobe is inactive which indicates that the microprocessor is not using the bus,
3. data transfer acknowledge is inactive which indicates that neither memory nor peripherals are using the bus, and
4. bus grant acknowledge is inactive which indicates that no other device is still claiming bus mastership.

4.1.5 Interrupt Control ($\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$)

These input pins indicate the encoded priority level of the device requesting an interrupt. Level seven is the highest priority while level zero indicates that no interrupts are requested. Level seven cannot be masked. The least significant bit is given in $\overline{IPL0}$ and the most significant bit is contained in $\overline{IPL2}$. These lines must remain stable until the processor signals interrupt acknowledge (FC0-FC2 are all high) to insure that the interrupt is recognized.

4.1.6 System Control

The system control inputs are used to either reset or halt the processor and to indicate to the processor that bus errors have occurred. The three system control inputs are explained in the following paragraphs.

4.1.6.1 BUS ERROR ($\overline{BERR}$). This input informs the processor that there is a problem with the cycle currently being executed. Problems may be a result of:

1. nonresponding devices,
2. interrupt vector number acquisition failure,
3. illegal access request as determined by a memory management unit, or
4. other application dependent errors.

The bus error signal interacts with the halt signal to determine if the current bus cycle should be re-executed or if exception processing should be performed.

Refer to **4.2.4 Bus Error and Halt Operation** for additional information about the interaction of the bus error and halt signals.

4.1.6.2 RESET ($\overline{RESET}$). This bidirectional signal line acts to reset (start a system initialization sequence) the processor in response to an external reset signal. An internally generated reset (result

of a $\overline{\text{RESET}}$ instruction) causes all external devices to be reset and the internal state of the processor is not affected. A total system reset (processor and external devices) is the result of external $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ signals applied at the same time. Refer to **4.2.5 Reset Operation** for further information.

4.1.6.3 HALT ($\overline{\text{HALT}}$). When this bidirectional line is driven by an external device, it will cause the processor to stop at the completion of the current bus cycle. When the processor has been halted using this input, all control signals are inactive and all three-state lines are put in their high-impedance state (refer to Table 4-3). Refer to **4.2.4 Bus Error and Halt Operation** for additional information about the interaction between the $\overline{\text{HALT}}$ and bus error signals.

When the processor has stopped executing instructions, such as in a double bus fault condition (refer to **4.2.4.4 DOUBLE BUS FAULTS**), the $\overline{\text{HALT}}$ line is driven by the processor to indicate to external devices that the processor has stopped.

4.1.7 EF6800 Peripheral Control

These control signals are used to allow the interfacing of synchronous EF6800 peripheral devices with the asynchronous TS68000. These signals are explained in the following paragraphs.

4.1.7.1 ENABLE (E). This signal is the standard enable signal common to all EF6800 type peripheral devices. The period for this output is ten TS68000 clock periods (six clocks low, four clocks high). Enable is generated by an internal ring counter which may come up in any state (i.e., at power on, it is impossible to guarantee phase relationship of E to CLK). E is a free-running clock and runs regardless of the state of the bus on the MPU.

4.1.7.2 VALID PERIPHERAL ADDRESS ($\overline{\text{VPA}}$). This input indicates that the device or region addressed is an EF6800 Family device and that data transfer should be synchronized with the enable (E) signal. This input also indicates that the processor should use automatic vectoring for an interrupt. Refer to **SECTION 6 INTERFACE WITH EF6800 PERIPHERALS.**

4.1.7.3 VALID MEMORY ADDRESS ($\overline{\text{VMA}}$). This output is used to indicate to EF6800 peripheral devices that there is a valid address on the address bus and the processor is synchronized to enable. This signal only responds to a valid peripheral address ($\overline{\text{VPA}}$) input which indicates that the peripheral is an EF6800 Family device.

4.1.8 Processor Status (FC0, FC1, FC2)

These function code outputs indicate the state (user or supervisor) and the cycle type currently being executed, as shown in Table 4-2. The information indicated by the function code outputs is valid whenever address strobe ($\overline{\text{AS}}$) is active.

Table 4-2. Function Code Outputs

Function Code Output			Cycle Type
FC2	FC1	FC0	
Low	Low	Low	(Undefined, Reserved)
Low	Low	High	User Data
Low	High	Low	User Program
Low	High	High	(Undefined, Reserved)

Function Code Output			Cycle Type
FC2	FC1	FC0	
High	Low	Low	(Undefined, Reserved)
High	Low	High	Supervisor Data
High	High	Low	Supervisor Program
High	High	High	Interrupt Acknowledge

4.1.9 Clock (CLK)

The clock input is a TTL-compatible signal that is internally buffered for development of the internal clocks needed by the processor. The clock input should not be gated off at any time and the clock signal must conform to minimum and maximum pulse width times.

4.1.10 Signal Summary

Table 4-3 is a summary of all the signals discussed in the previous paragraphs.

Table 4-3. Signal Summary

Signal Name	Mnemonic	Input/Output	Active State	Hi-Z	
				On $\overline{HALT}$	On $\overline{BGACK}$
Address Bus	A1-A23	Output	High	Yes	Yes
Data Bus	D0-D15	Input Output	High	Yes	Yes
Address Strobe	$\overline{AS}$	Output	Low	No	Yes
Read/Write	R/$\overline{W}$	Output	Read-High Write-Low	No	Yes
Upper and Lower Data Stobes	$\overline{UDS}$, $\overline{LDS}$	Output	Low	No	Yes
Data Transfer Acknowledge	$\overline{DTACK}$	Input	Low	No	No
Bus Request	$\overline{BR}$	Input	Low	No	No
Bus Grant	$\overline{BG}$	Output	Low	No	No
Bus Grant Acknowledge	$\overline{BGACK}$	Input	Low	No	No
Interrupt Priority Level	$\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$	Input	Low	No	No
Bus Error	$\overline{BERR}$	Input	Low	No	No
Reset	$\overline{RESET}$	Input Output	Low	No[1]	No[1]
Halt	$\overline{HALT}$	Input Output	Low	No[1]	No[1]
Enable	E	Output	High	No	No
Valid Memory Address	$\overline{VMA}$	Output	Low	No	Yes
Valid Peripheral Address	$\overline{VPA}$	Input	Low	No	No
Function Code Output	FC0, FC1, FC2	Output	High	No	Yes
Clock	CLK	Input	High	No	No
Power Input	V_{CC}	Input	-	-	-
Ground	GND	Input	-	-	-

NOTE :
1 Open drain

4.2 BUS OPERATION

The following paragraphs explain control signal and bus operation during data transfer operations, bus arbitration, bus error and halt conditions, and reset operation.

4.2.1 Data Transfer Operations

Transfer of data between devices involves the following leads:

1. address bus A1 through A23,
2. data bus D0 through D15, and
3. control signals.

The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure. In all cycles, the bus master assumes responsibility for deskewing all signals it issues at both the start and end of a cycle. In addition, the bus master is responsible for deskewing the acknowledge and data signals from the slave device.

The following paragraphs explain the read, write, and read-modify-write cycles. The indivisible read-modify-write cycle is the method used by the TS68000 for interlocked multiprocessor communications.

4.2.1.1 READ CYCLE. During a read cycle, the processor receives data from the memory or a peripheral device. The processor reads bytes of data in all cases. If the instruction specifies a word (or double word) operation, the processor reads both upper and lower bytes simultaneously by asserting both upper and lower data strobes. When the instruction specifies byte operation, the processor uses an internal A0 bit to determine which byte to read and then issues the data strobe required for that byte. For byte operations, when the A0 bit equals zero, the upper data strobe is issued. When the A0 bit equals one, the lower data strobe is issued. When the data is received, the processor correctly positions it internally.

A word read cycle flowchart is given in Figure 4-2. A byte read cycle flowchart is given in Figure 4-3. Read cycle timing is given in Figure 4-4. Figure 4-5 details word and byte read cycle operations.

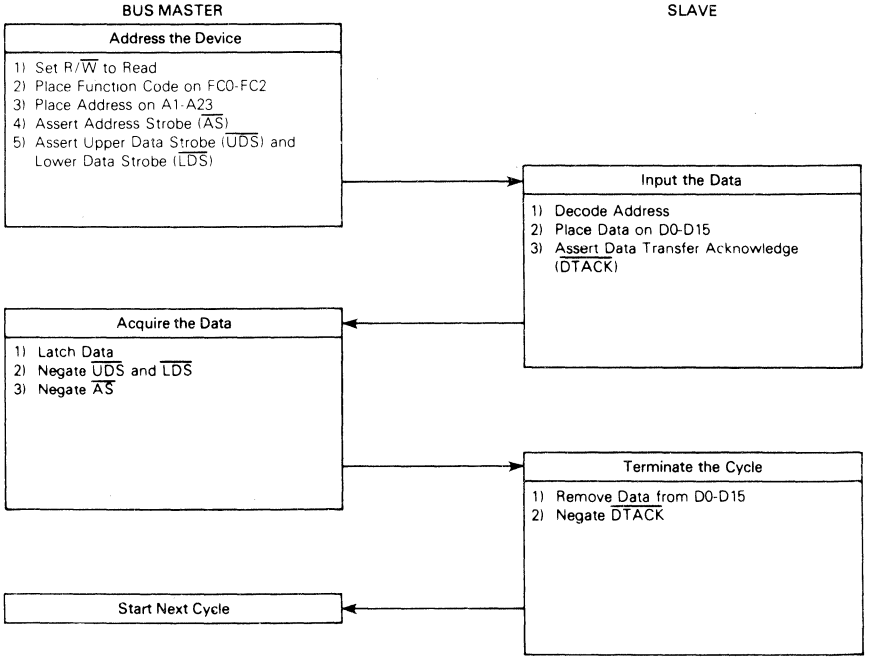


Figure 4-2. Word Read Cycle Flowchart

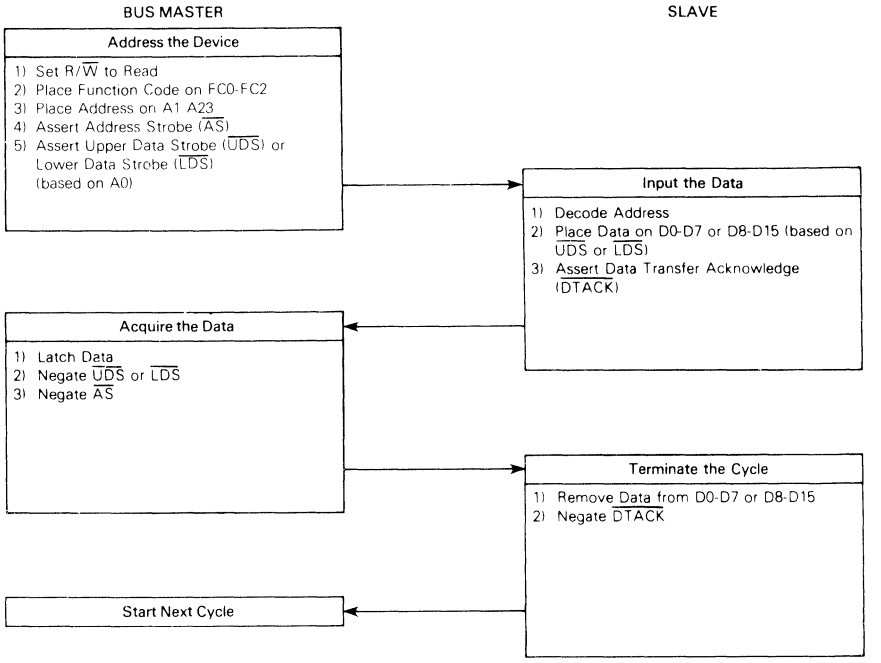


Figure 4-3. Byte Read Cycle Flowchart

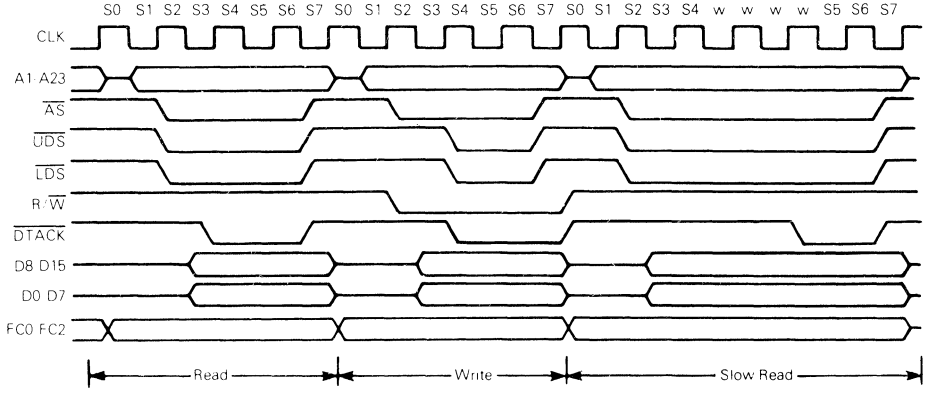


Figure 4-4. Read and Write Cycle Timing Diagram

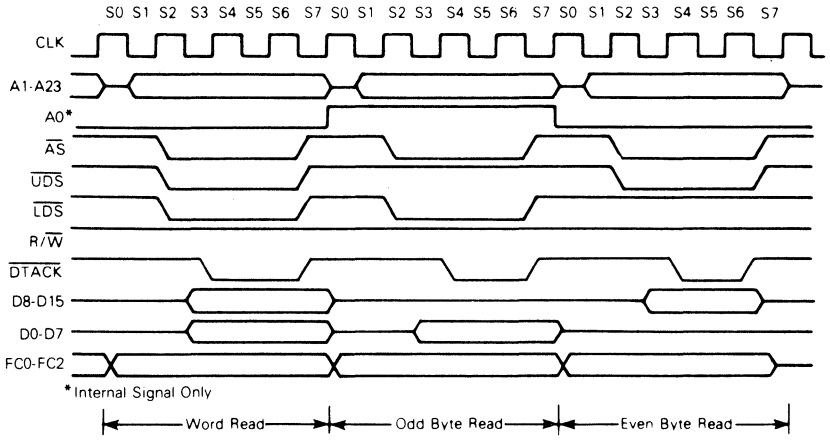


Figure 4-5. Word and Byte Read Cycle Timing Diagram

4.2.1.2 WRITE CYCLE. During a write cycle, the processor sends data to either the memory or a peripheral device. The processor writes bytes of data in all cases. If the instruction specifies a word operation, the processor writes both bytes. When the instruction specifies a byte operation, the processor uses an internal A0 bit to determine which byte to write and then issues the data strobe required for that byte. For byte operations, when the A0 bit equals zero, the upper data strobe is issued. When the A0 bit equals one, the lower data strobe is issued. A word write flowchart is given in Figure 4-6. A byte write cycle flowchart is given in Figure 4-7. Write cycle timing is given in Figure 4-4. Figure 4-8 details word and byte write cycle operation.

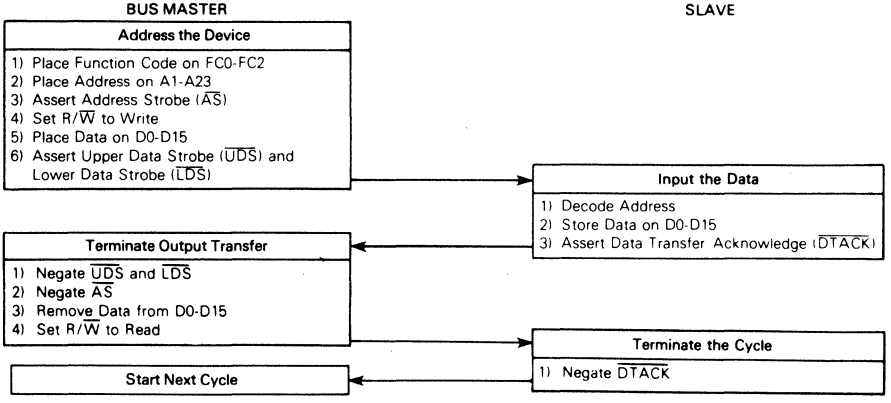


Figure 4-6. Word Write Cycle Flowchart

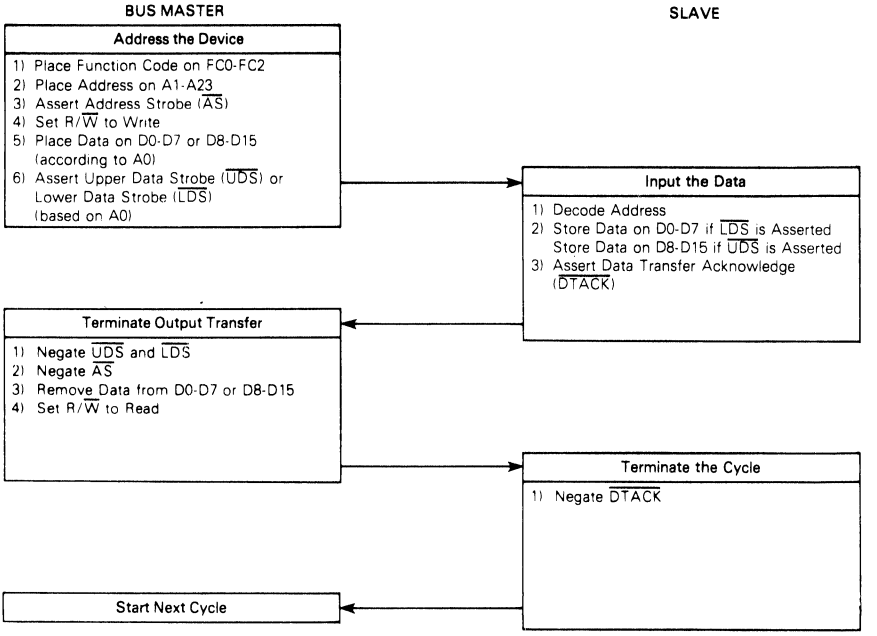


Figure 4-7. Byte Write Cycle Flowchart

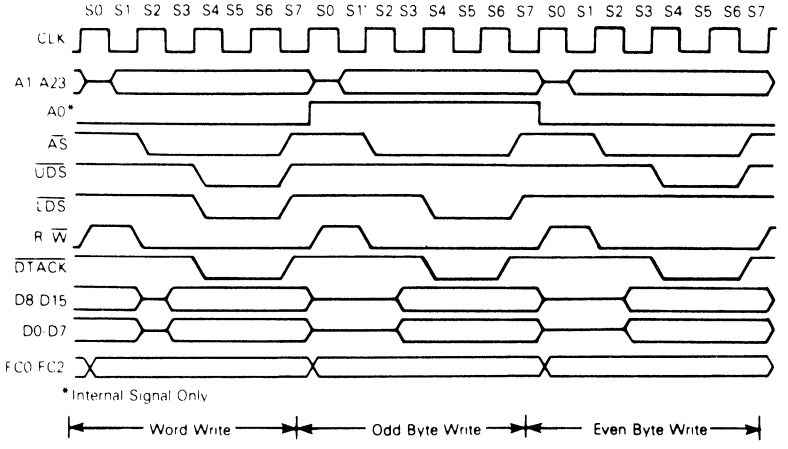


Figure 4-8. Word and Byte Write Cycle Timing Diagram

4.2.1.3 READ-MODIFY-WRITE CYCLE. The read-modify-write cycle performs a read, modifies the data in the arithmetic-logic unit, and writes the data back to the same address. In the TS68000, this cycle is indivisible in that the address strobe is asserted throughout the entire cycle. The test and set (TAS) instruction uses this cycle to provide meaningful communication between processors in a multiple processor environment. This instruction is the only instruction that uses the read-modify-write cycles and since the test and set instruction only operates on bytes, all read-modify-write cycles are byte operations. A read-modify-write flowchart is given in Figure 4-9 and a timing diagram is given in Figure 4-10.

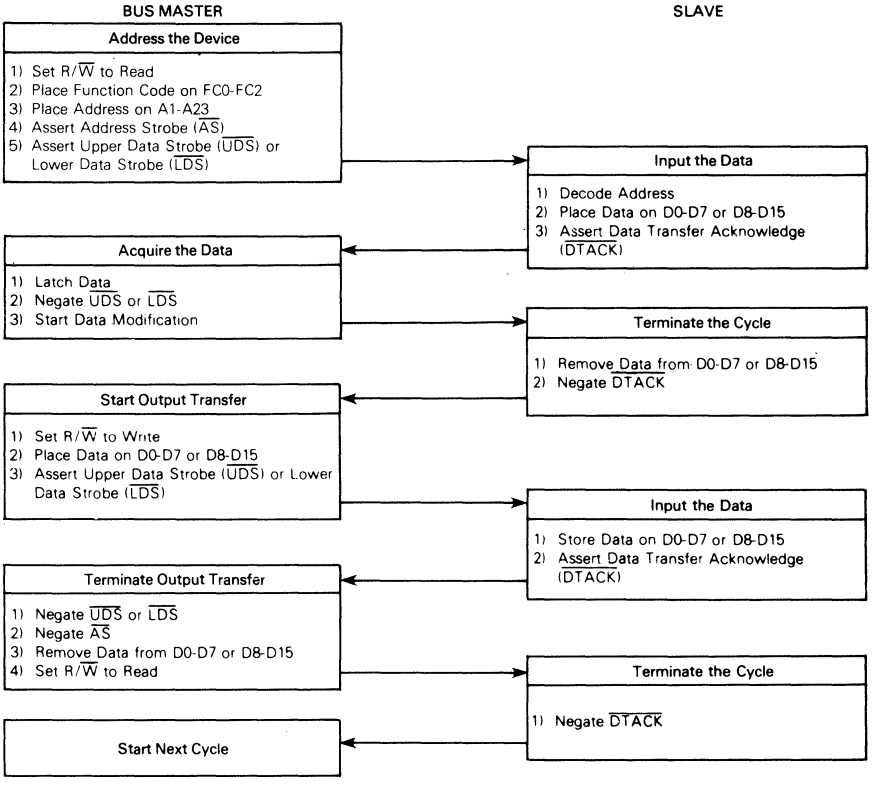


Figure 4-9. Read-Modify-Write Cycle Flowchart

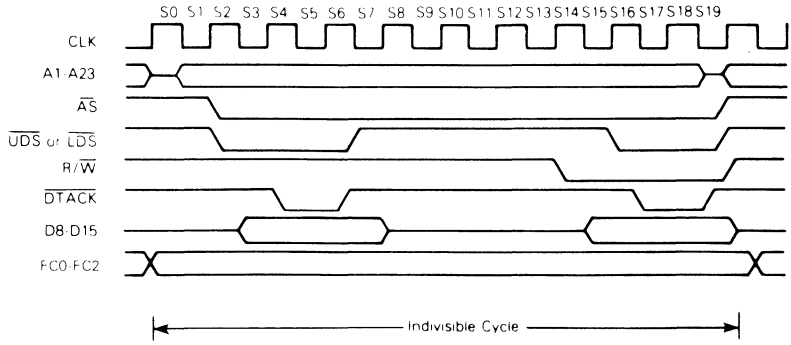


Figure 4-10. Read-Modify-Write Cycle Timing Diagram

4.2.2 Bus Arbitration

Bus arbitration is a technique used by master-type devices to request, be granted, and acknowledge bus mastership. In its simplest form, it consists of the following:

1. asserting a bus mastership request,
2. receiving a grant that the bus is available at the end of the current cycle, and
3. acknowledging that mastership has been assumed.

Figure 4-11 is a flowchart showing the detail involved in a request from a single device. Figure 4-12 is a timing diagram for the same operation. This technique allows processing of bus requests during data transfer cycles.

The timing diagram shows that the bus request is negated at the time that an acknowledge is asserted. This type of operation would be true for a system consisting of the processor and one device capable of bus mastership. In systems having a number of devices capable of bus mastership, the bus request line from each device is wire ORed to the processor. In this system, it is easy to see that there could be more than one bus request being made. The timing diagram shows that the bus grant signal is negated a few clock cycles after the transition of the acknowledge ($\overline{BGACK}$) signal.

However, if the bus requests are still pending, the processor will assert another bus grant within a few clock cycles after it was negated. This additional assertion of bus grant allows external arbitration circuitry to select the next bus master before the current bus master has completed its requirements. The following paragraphs provide additional information about the three steps in the arbitration process.

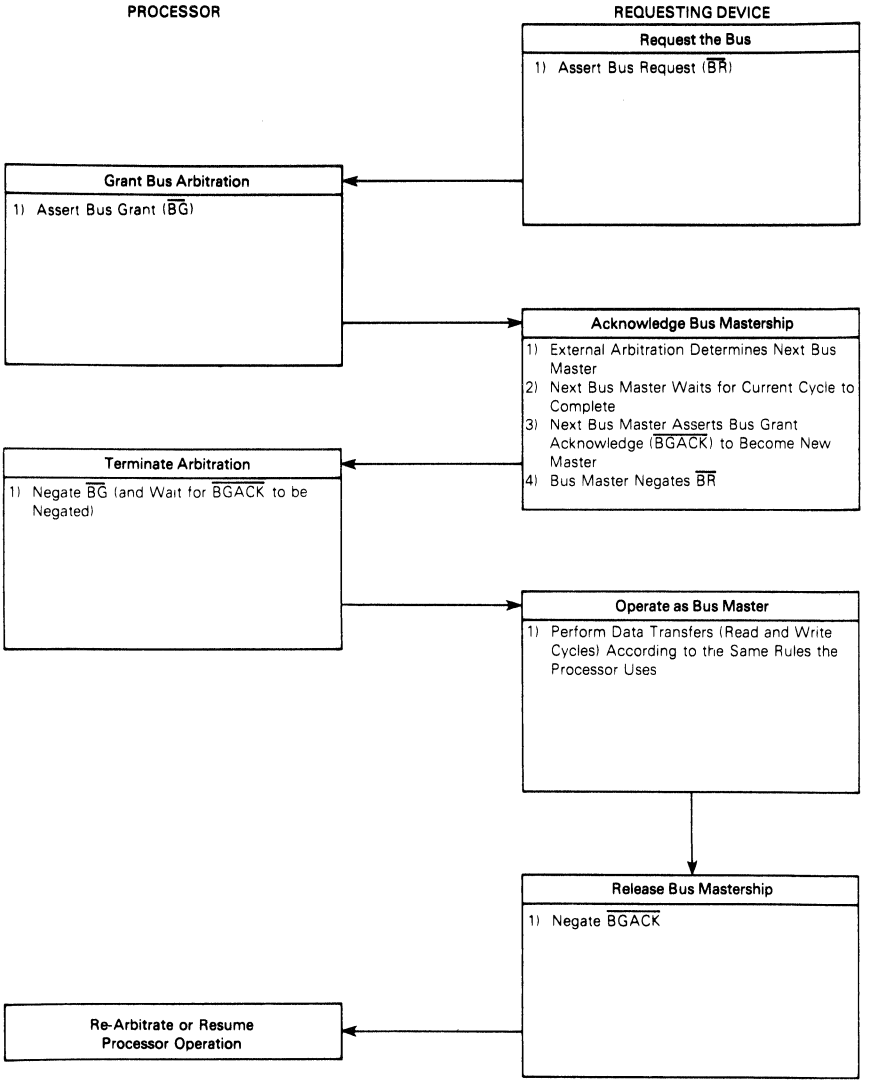


Figure 4-11. Bus Arbitration Cycle Flowchart

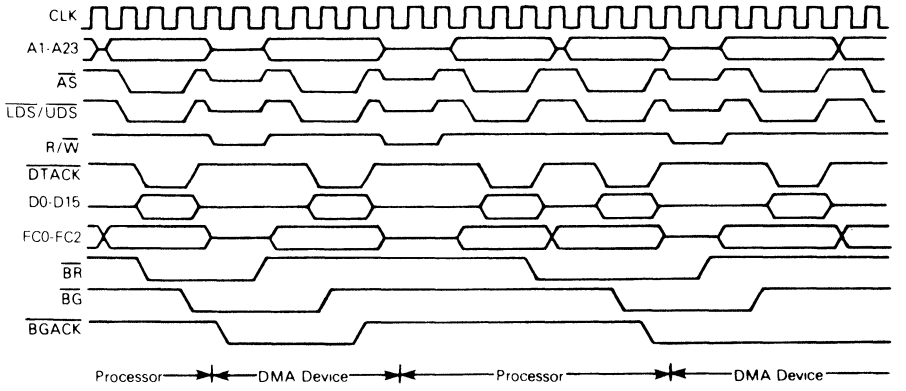


Figure 4-12. Bus Arbitration Cycle Timing Diagram

4.2.2.1 REQUESTING THE BUS. External devices capable of becoming bus masters request the bus by asserting the bus request ($\overline{BR}$) signal. This is a wire-ORed signal (although it need not be constructed from open-collector devices) that indicates to the processor that some external device requires control of the external bus. The processor is effectively at a lower bus priority level than the external device and will relinquish the bus after it has completed the last bus cycle it has started.

When no acknowledge is received before the bus request signal goes inactive, the processor will continue processing when it detects that the bus request is inactive. This allows ordinary processing to continue if the arbitration circuitry responded to noise inadvertently.

4.2.2.2 RECEIVING THE BUS GRANT. The processor asserts bus grant ($\overline{BG}$) as soon as possible. Normally this is immediately after internal synchronization. The only exception to this occurs when the processor has made an internal decision to execute the next bus cycle but has not progressed far enough into the cycle to have asserted the address strobe ($\overline{AS}$) signal. In this case, bus grant will be delayed until $\overline{AS}$ is asserted to indicate to external devices that a bus cycle is being executed.

The bus grant signal may be routed through a daisy-chained network or through a specific priority-encoded network. The processor is not affected by the external method of arbitration as long as the protocol is obeyed.

4.2.2.3 ACKNOWLEDGEMENT OF MASTERSHIP. Upon receiving a bus grant, the requesting device waits until address strobe, data transfer acknowledge, and bus grant acknowledge are negated before issuing its own $\overline{BGACK}$. The negation of the address strobe indicates that the previous master has completed its cycle; the negation of bus grant acknowledge indicates that the previous master has released the bus. (While address strobe is asserted, no device is allowed to "break into" a cycle.) The negation of data transfer acknowledge indicates the previous slave has terminated its connection to the previous master. Note that in some applications data transfer acknowledge might not enter into this function. General purpose devices would then be connected such that they were only dependent on address strobe. When bus grant acknowledge is issued, the

device is a bus master until it negates bus grant acknowledge. Bus grant acknowledge should not be negated until after the bus cycle(s) is (are) completed. Bus mastership is terminated at the negation of bus grant acknowledge.

The bus request from the granted device should be dropped after bus grant acknowledge is asserted. If a bus request is still pending, another bus grant will be asserted within a few clocks of the negation of the bus grant. Refer to **4.2.3 Bus Arbitration Control.** Note that the processor does not perform any external bus cycles before it re-asserts bus grant.

4.2.3 Bus Arbitration Control

The bus arbitration control unit in the TS68000 is implemented with a finite state machine. A state diagram of this machine is shown in Figure 4-13. All asynchronous signals to the TS68000 are synchronized before being used internally. This synchronization is accomplished in a maximum of one cycle of the system clock, assuming that the asynchronous input setup time (#47) has been met (see Figure 4-14). The input signal is sampled on the falling edge of the clock and is valid internally after the next falling edge.

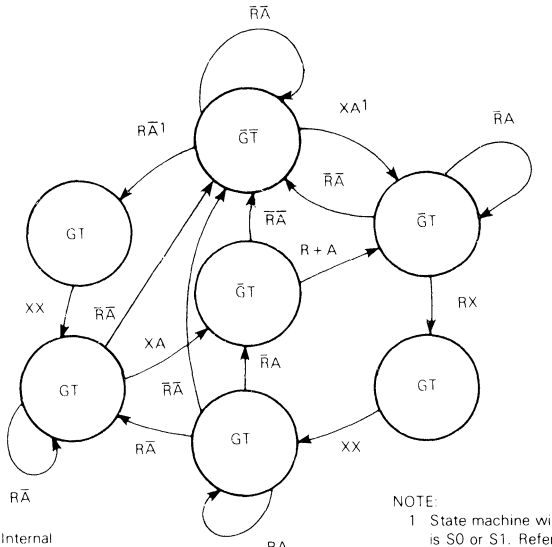
As shown in Figure 4-13, input signals labeled R and A are internally synchronized on the bus request and bus grant acknowledge pins respectively. The bus grant output is labeled G and the internal three-state control signal T. If T is true, the address, data, and control buses are placed in a high-impedance state when $\overline{AS}$ is negated. All signals are shown in positive logic (active high) regardless of their true active voltage level. State changes (valid outputs) occur on the next rising edge after the internal signal is valid.

A timing diagram of the bus arbitration sequence during a processor bus cycle is shown in Figure 4-15. The bus arbitration sequence while the bus is inactive (i.e., executing internal operations such as a multiply instruction) is show in Figure 4-16.

If a bus request is made at a time when the MPU has already begun a bus cycle but $\overline{AS}$ has not been asserted (bus state S0), $\overline{BG}$ will not be asserted on the next rising edge. Instead, $\overline{BG}$ will be delayed until the second rising edge following its internal assertion. This sequence is shown in Figure 4-17.

4.2.4 Bus Error and Halt Operation

In a bus architecture that requires a handshake from an external device, the possibility exists that the handshake might not occur. Since different systems will require a different maximum response time, a bus error input is provided. External circuitry must be used to determine the duration between address strobe and data transfer acknowledge before issuing a bus error signal. When a bus error signal is received, the processor has two options: initiate a bus error exception sequence or try running the bus cycle again.



R = Bus Reqest Internal
A = Bus Grant Acknowledge Internal
G = Bus Grant
T = Three State Control to Bus Control Logic[2]
X = Don't Care

NOTE:
1. State machine will not change if the bus is S0 or S1. Refer to **4.2.3 Bus Arbitration Contol.**
2. The address bus will be placed in the high-impedance state if T is asserted and $\overline{AS}$ is negated.

Figure 4-13. TS68000 Bus Arbitration Unit State Diagram

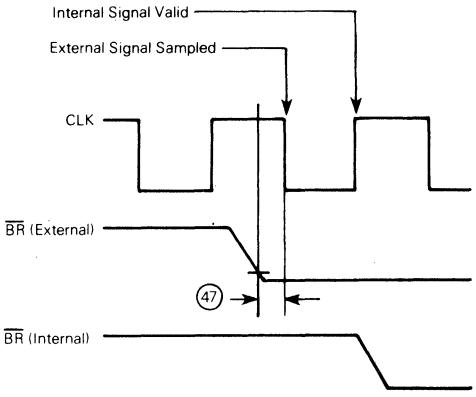


Figure 4-14. Timing Relationship of External Asynchronous Inputs to Internal Signals

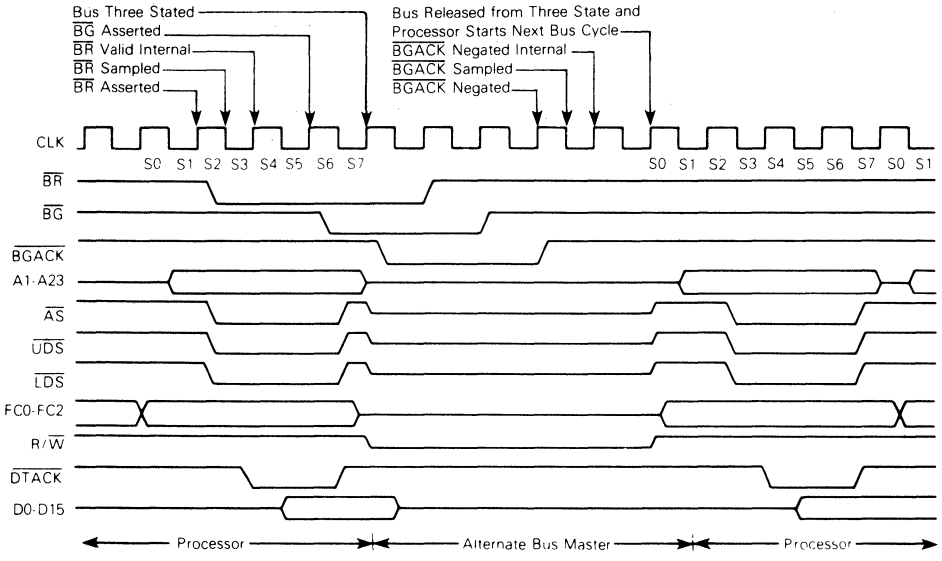


Figure 4-15. Bus Arbitration Timing Diagram — Processor Active

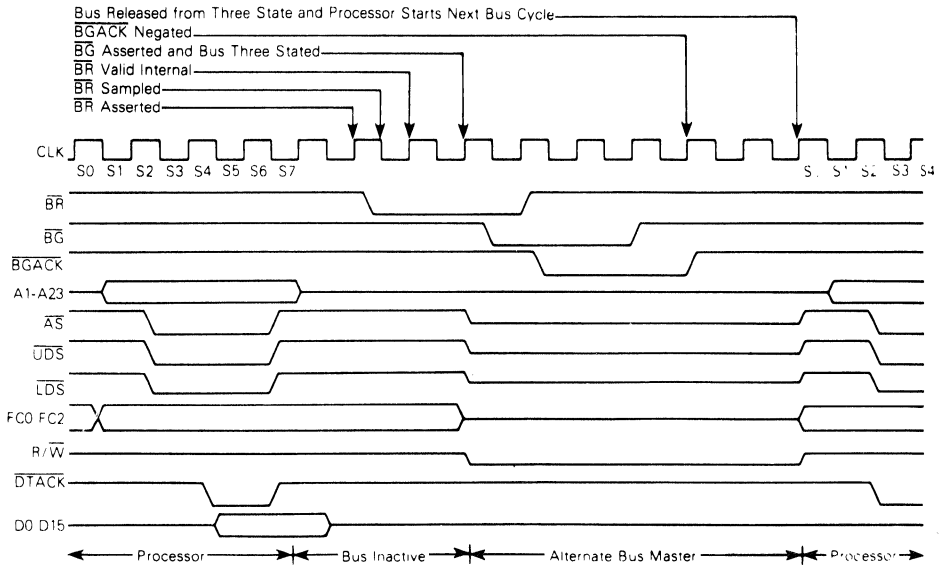


Figure 4-16. Bus Arbitration Timing Diagram — Bus Inactive

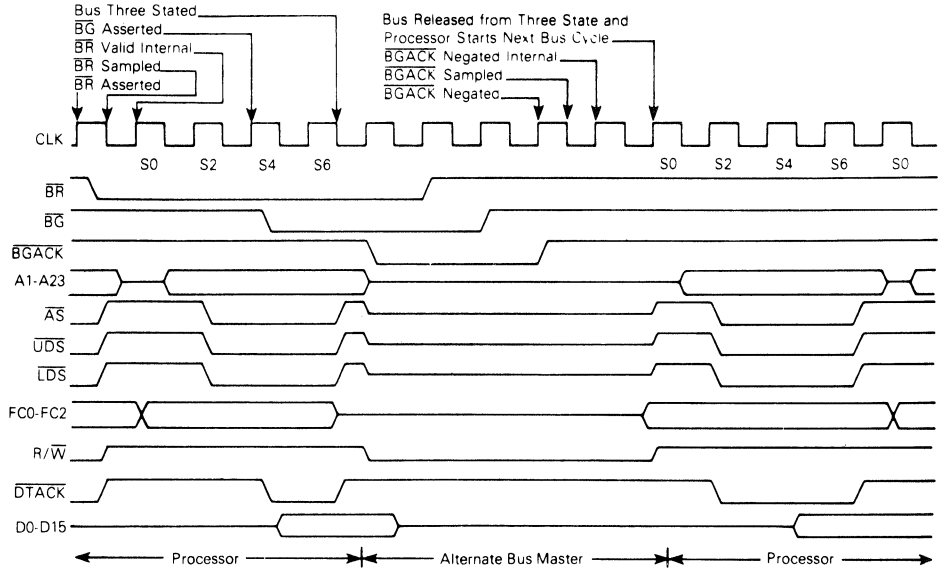


Figure 4-17. Bus Arbitration Timing Diagram — Special Case

4.2.4.1 BUS ERROR OPERATION. When the bus error signal is asserted, the current bus cycle is terminated. If $\overline{BERR}$ is asserted before the falling edge of S2, $\overline{AS}$ will be negated in S7 in either a read or write cycle. As long as $\overline{BERR}$ remains asserted, the data and address buses will be in the high-impedence state. When $\overline{BERR}$ is negated, the processor will begin stacking for exception processing. Figure 4-18 is a timing diagram for the exception sequence. The sequence is composed of the following elements:

1. stacking the program counter and status register,
2. stacking the error information,
3. reading the bus error vector table entry, and
4. executing the bus error handler routine.

The stacking of the program counter and the status register is the same as if an interrupt had occurred. Several additional items are stacked when a bus error occurs. These items are used to determine the nature of the error and correct it, if possible. The bus error vector is vector number two located at address $000008. The processor loads the new program counter from this location. A software bus error handler routine is then executed by the processor. Refer to **5.2 EXCEPTION PROCESSING** for additional information.

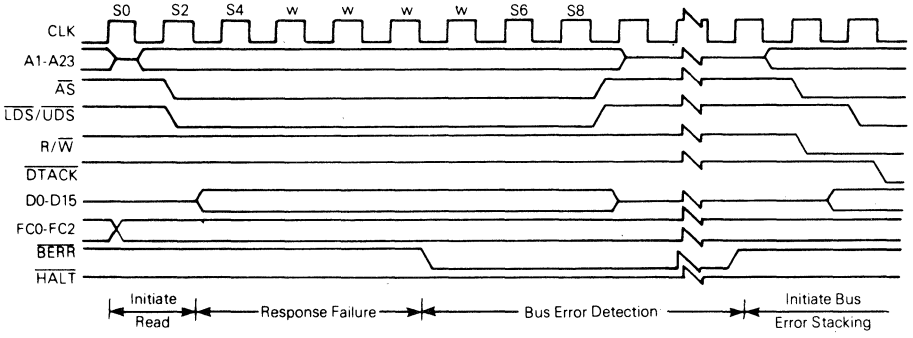


Figure 4-18. Bus Error Timing Diagram

4.2.4.2 RE-RUN OPERATION. When, during a bus cycle, the processor receives a bus error signal and the halt pin is being driven by an external device, the processor enters the re-run sequence. Figure 4-19 is a timing diagram for re-running the bus cycle.

The processor terminates the bus cycle, then puts the address and data output lines in the high-impedence state. The processor remains "halted", and will not run another bus cycle until the halt signal is removed by external logic. Then the processor will re-run the previous cycle using the same function codes, the same data (for a write operation), and the same controls. The bus error signal should be removed at least one clock cycle before the halt signal is removed.

NOTE

The processor will not re-run a read-modify-write cycle. This restriction is made to guarantee that the entire cycle runs correctly and that the write operation of a test-and-set operation is performed without ever releasing $\overline{AS}$. If $\overline{BERR}$ and $\overline{HALT}$ are asserted during a read-modify-write bus cycle, a bus error operation results.

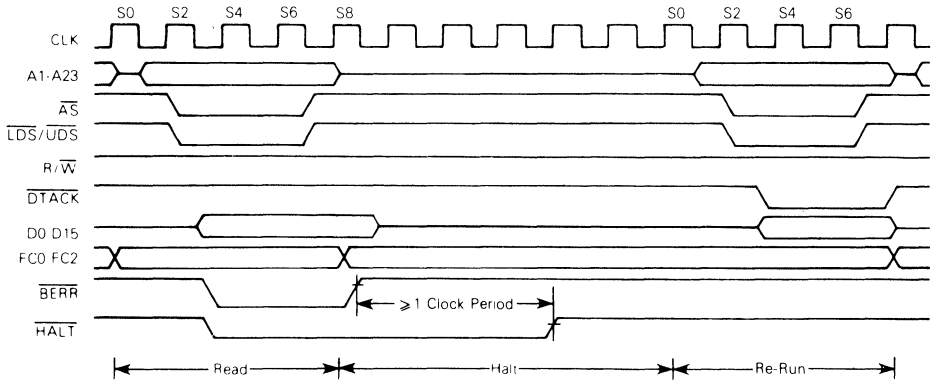


Figure 4-19. Re-Run Bus Cycle Timing Diagram

4.2.4.3 HALT OPERATION. The halt input signal to the TS68000 performs a halt/run/single-step function in a similar fashion to the EF6800 halt function. The halt and run modes are somewhat self explanatory in that when the halt signal is constantly active the processor "halts" (does nothing) and when the halt signal is constantly inactive the processor "runs" (does something).

This single-step mode is derived from correctly timed transitions on the halt signal input. It forces the processor to execute a single bus cycle by entering the run mode until the processor starts a bus cycle then changing to the halt mode. Thus, the single-step mode allows the user to proceed through (and therefore debug) processor operations one bus cycle at a time.

Figure 4-20 details the timing required for correct single-step operations. Some care must be exercised to avoid harmful interactions between the bus error signal and the halt pin when using the single-cycle mode as a debugging tool. This is also true of interactions between the halt and reset lines since these can reset the machine.

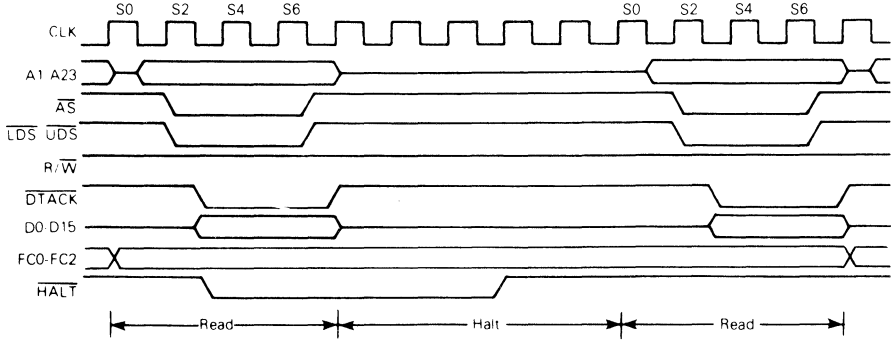


Figure 4-20. Halt Processor Timing Diagram

When the processor completes a bus cycle after recognizing that the halt signal is active, most three-state signals are put in the high-impedence state, these include:

1. address lines, and
2. data lines.

This is required for correct performance of the re-run bus cycle operation.

While the processor is honoring the halt request, bus arbitration performs as usual. That is, halting has no effect on bus arbitration. It is the bus arbitration function that removes the control signals from the bus.

The halt function and the hardware trace capability allow the hardware debugger to trace single bus cycles or single instructions at a time. These processor capabilities, along with a software debugging package, give total debugging flexibility.

4.2.4.4 DOUBLE BUS FAULTS. When a bus error exception occurs, the processor will attempt to stack several words containing information about the state of the machine. If a bus error exception occurs during the stacking operation, there have been two bus errors in a row. This is commonly referred to as a double bus fault. When a double bus fault occurs, the processor will halt. Once a bus error exception has occurred, any bus error exception occurring before the execution of the next instruction constitutes a double bus fault.

Note that a bus cycle which is re-run does not constitute a bus error exception and does not contribute to a double bus fault. Note also that this means that as long as the external hardware requests it, the processor will continue to re-run the same bus cycle.

The bus error pin also has an effect on processor operation after the processor receives an external reset input. The processor reads the vector table after a reset to determine the address to start program execution. If a bus error occurs while reading the vector table (or at any time before the first instruction is executed), the processor reacts as if a double bus fault has occurred and it halts. Only an external reset will start a halted processor.

4.2.5 Reset Operation

The reset signal is a bidirectional signal that allows either the processor or an external signal to reset the system. Figure 4-21 is a timing diagram for the reset operation. Both the halt and reset lines must be asserted to ensure total reset of the processor.

When the reset and halt lines are driven by an external device, it is recognized as an entire system reset, including the processor. The processor responds by reading the reset vector table entry (vector number zero, address $000000) and loads it into the supervisor stack pointer (SSP). Vector table entry number one at address $000004 is read next and loaded into the program counter. The processor initializes the status register to an interrupt level of seven. No other registers are affected by the reset sequence.

When a reset instruction is executed, the processor drives the reset pin for 124 clock periods. In this case, the processor is trying to reset the rest of the system. Therefore, there is no effect on the

internal state of the processor. All of the processor's internal registers and the status register are unaffected by the execution of a reset instruction. All external devices connected to the reset line will be reset at the completion of the reset instruction.

Asserting the reset and halt lines for ten clock cycles will cause a processor reset, except when V_{CC} is initially applied to the processor. In this case, an external reset must be applied for at least 100 milliseconds.

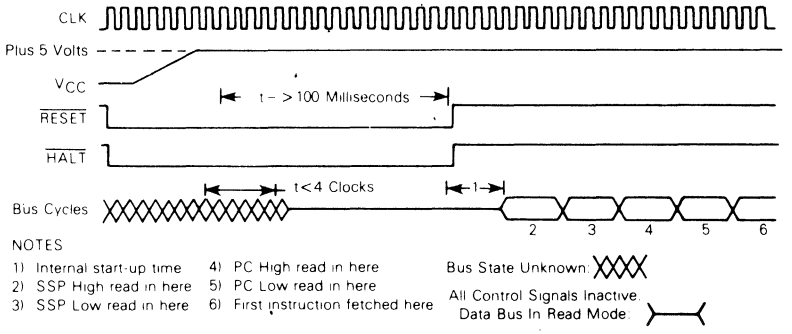


Figure 4-21. Reset Operation Timing Diagram

4.3 THE RELATIONSHIP OF $\overline{DTACK}$, $\overline{BERR}$, AND $\overline{HALT}$

In order to properly control termination of a bus cycle for a re-run or a bus error condition, $\overline{DTACK}$, $\overline{BERR}$, and $\overline{HALT}$ should be asserted and negated on the rising edge of the TS68000 clock. This will assure that when two signals are asserted simultaneously, the required setup time (#47) for both of them will be met during the same bus state.

This, or some equivalent precaution, should be designed external to the TS68000. Parameter #48 is intended to ensure this operation in a totally asynchronous system, and may be ignored if the above conditions are met.

The preferred bus cycle terminations may be summarized as follows (case numbers refer to Table 4-4):

Normal Termination:	$\overline{DTACK}$ occurs first (case 1).
Halt Termination:	$\overline{HALT}$ is asserted at the same time or before $\overline{DTACK}$ and $\overline{BERR}$ remains negated (cases 2 and 3).
Bus Error Termination:	$\overline{BERR}$ is asserted in lieu of, at the same time, or before $\overline{DTACK}$ (case 4); $\overline{BERR}$ is negated at the same time or after $\overline{DTACK}$.
Re-Run Termination:	$\overline{HALT}$ and $\overline{BERR}$ are asserted in lieu of, at the same time, or before $\overline{DTACK}$ (cases 6 and 7); $\overline{HALT}$ must be held at least one cycle after $\overline{BERR}$. Case 5 indicates $\overline{BERR}$ may precede $\overline{HALT}$ on all mask sets which allows fully asynchronous assertion.

Table 4-4 details the resulting bus cycle termination under various combinations of control signal sequences. The negation of these same control signals under several conditions is shown in Table 4-5 ($\overline{DTACK}$ is assumed to be negated normally in all cases; for best results, both $\overline{DTACK}$ and $\overline{BERR}$ should be negated when address strobe is negated).

Table 4-4. $\overline{DTACK}$, $\overline{BERR}$, and $\overline{HALT}$ Assertion Results

Case No.	Control Signal	Asserted on Rising Edge of State		Result
		N	N+2	
1	$\overline{DTACK}$	A	S	Normal cycle terminate and continue.
	$\overline{BERR}$	NA	X	
	$\overline{HALT}$	NA	X	
2	$\overline{DTACK}$	A	S	Normal cycle terminate and halt. Continue when $\overline{HALT}$ removed.
	$\overline{BERR}$	NA	X	
	$\overline{HALT}$	A	S	
3	$\overline{DTACK}$	NA	A	Normal cycle terminate and halt. Continue when $\overline{HALT}$ removed.
	$\overline{BERR}$	NA	NA	
	$\overline{HALT}$	A	S	
4	$\overline{DTACK}$	X	X	Terminate and take bus error trap.
	$\overline{BERR}$	A	S	
	$\overline{HALT}$	NA	NA	
5	$\overline{DTACK}$	NA	X	Terminate and re-run.
	$\overline{BERR}$	A	S	
	$\overline{HALT}$	NA	A	
6	$\overline{DTACK}$	X	X	Terminate and re-run when $\overline{HALT}$ removed.
	$\overline{BERR}$	A	S	
	$\overline{HALT}$	A	S	
7	$\overline{DTACK}$	NA	X	Terminate and re-run when $\overline{HALT}$ removed.
	$\overline{BERR}$	NA	A	
	$\overline{HALT}$	A	S	

Legend:
N – the number of the current even bus state (e.g., S4, S6, etc.)
A – signal is asserted in this bus state
NA – signal is not asserted in this state
X – don't care
S – signal was asserted in previous state and remains asserted in this state

Table 4-5. $\overline{BERR}$ and $\overline{HALT}$ Negation Results

Conditions of Termination in Table 4-4	Control Signal	Negated on Rising Edge of State			Results – Next Cycle
		N		N+2	
Bus Error	$\overline{BERR}$	●	or	●	Takes bus error trap.
	$\overline{HALT}$	●	or	●	
Re-run	$\overline{BERR}$	●	or	●	Illegal sequence; usually traps to vector number 0.
	$\overline{HALT}$	●			
Re-run	$\overline{BERR}$	●			Re-runs the bus cycle.
	$\overline{HALT}$			●	
Normal	$\overline{BERR}$	●			May lengthen next cycle.
	$\overline{HALT}$	●	or	●	
Normal	$\overline{BERR}$			●	If next cycle is started it will be terminated as a bus error.
	$\overline{HALT}$	●	or	none	

● = Signal is negated in this bus state.

EXAMPLE A:

A system uses a watch-dog timer to terminate accesses to unpopulated address space. The timer asserts $\overline{DTACK}$ and $\overline{BERR}$ simultaneously after time out (case 4).

EXAMPLE B:

A system uses error detection on RAM contents. Designer may (a) delay $\overline{DTACK}$ until data verified and return $\overline{BERR}$ and $\overline{HALT}$ simultaneously to re-run error cycle (case 6), or if valid, return $\overline{DTACK}$ (case 1); (b) delay $\overline{DTACK}$ until data verified and return $\overline{BERR}$ at same time as $\overline{DTACK}$ if data in error (case 4).

4.4 ASYNCHRONOUS VERSUS SYNCHRONOUS OPERATION

4.4.1 Asynchronous Operation

To achieve clock frequency independence at a system level, the TS68000 can be used in an asynchronous manner. This entails using only the bus handshake lines ($\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, $\overline{DTACK}$, $\overline{BERR}$, $\overline{HALT}$, and $\overline{VPA}$) to control the data transfer. Using this method, $\overline{AS}$ signals the start of a bus cycle and the data strobes are used as a condition for valid data on a write cycle. The slave device (memory or peripheral) then responds by placing the requested data on the data bus for a read cycle or latching data on a write cycle and asserting the data transfer acknowlege signal ($\overline{DTACK}$) to terminate the bus cycle. If no slave responds or the access is invalid, external control logic asserts the $\overline{BERR}$, or $\overline{BERR}$ and $\overline{HALT}$, signal to abort or rerun the bus cycle.

The $\overline{DTACK}$ signal is allowed to be asserted before the data from a slave device is valid on a read cycle. The length of time that $\overline{DTACK}$ may precede data is given as parameter #31 and it must be met in any asynchronous system to insure that valid data is latched into the processor. Notice that there is no maximum time specified from the assertion of $\overline{AS}$ to the assertion of $\overline{DTACK}$. This is because the MPU will insert wait cycles of one clock period each until $\overline{DTACK}$ is recognized.

4.4.2 Synchronous Operation

To allow for those systems which use the system clock as a signal to generate $\overline{DTACK}$ and other asynchronous inputs, the asynchronous input setup time is given as parameter #47. If this setup is met on an input, such as $\overline{DTACK}$, the processor is guaranteed to recognize that signal on the next falling edge of the system clock. However, the converse is not true – if the input signal does not meet the setup time it is not guaranteed not to be recognized. In addition, if $\overline{DTACK}$ is recognized on a falling edge, valid data will be latched into the processor (on a read cycle) on the next falling edge provided that the data meets the setup time given as parameter #27. Given this, parameter #31 may be ignored. Note that if $\overline{DTACK}$ is asserted, with the required setup time, before the falling edge of S4, no wait states will be incurred and the bus cycle will run at its maximum speed of four clock periods.

NOTE

During an active bus cycle, $\overline{BERR}$ is sampled on every falling edge of the clock starting with S2. $\overline{DTACK}$ is sampled on every falling edge of the clock starting with S4 and data is latched on the falling edge of S6 during a read. The bus cycle will then be terminated in S7 except when $\overline{BERR}$ is asserted in the absence of $\overline{DTACK}$, in which case it will terminate one clock cycle later in S9. $\overline{VPA}$ is sampled only on the third falling edge of the system clock before the rising edge of the E clock.

SECTION 5
PROCESSING STATES

This section describes the actions of the TS68000 which are outside the normal processing associated with the execution of instructions. The functions of the bits in the supervisor portion of the status register are covered: the supervisor/user bit, the trace enable bit, and the processor interrupt priority mask. Finally, the sequence of memory references and actions taken by the processor on exception conditions are detailed.

The TS68000 is always in one of three processing states : normal, exception, or halted. The normal processing state is that associated with instruction execution; the memory references are to fetch instructions and operands, and to store results. A special case of the normal state is the stopped state which the processor enters when a stop instruction is executed. In this state, no further references are made.

The exception processing state is associated with interrupts, trap instructions, tracing, and other exceptional conditions. The exception may be internally generated by an instruction or by an unusual condition arising during the execution of an instruction. Externally, exception processing can be forced by an interrupt, by a bus error, or by a reset. Exception processing is design
vide an efficient context switch so that the processor may handle unusual conditions.

The halted processing state is an indication of catastrophic hardware failure. For example, if during the exception processing of a bus error another bus error occurs, the processor assumes that the system is unusable and halts. Only an external reset can restart a halted processor. Note that a processor in the stopped state is not in the halted state, nor vice versa.

5.1 PRIVILEGE STATES

The processor operates in one of two states of privilege: the "supervisor" state or the "user" state. The privilege state determines which operations are legal, are used to choose between the supervisor stack pointer and the user stack pointer in instruction references, and may by used by an external memory management device to control and translate accesses.

The privilege state is a mechanism for providing security in a computer system. Programs should access only their own code and data areas, and ought to be restricted from accessing information which they do not need and must not modify.

The privilege mechanism provides security by allowing most programs to execute in user state. In this state, the accesses are controlled, and the effects on other parts of the system are limited. The operating system executes in the supervisor state, has access to all resources, and performs the overhead tasks for the user state programs.

5.1.1 Supervisor State

The supervisor state is the higher state of privilege. For instruction execution, the supervisor state is determined by the S bit of the status register; if the S bit is asserted (high), the processor is in the supervisor state. All instructions can be executed in the supervisor state. The bus cycles generated by instructions executed in the supervisor state are classified as supervisor references. While the processor is in the supervisor privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly access the supervisor stack pointer.

All exception processing is done in the supervisor state, regardless of the setting of the S bit. The bus cycles generated during exception processing are classified as supervisor references. All stacking operations during exception processing use the supervisor stack pointer.

5.1.2 User State

The user state is the lower state of privilege. For instruction execution, the user state is determined by the S bit of the status register; if the S bit is negated (low), the processor is executing instructions in the user state.

Most instructions execute the same in user state as in the supervisor state. However, some instructions which have important system effects are made privileged. User programs are not permitted to execute the stop instruction or the reset instruction. To ensure that a user program cannot enter the supervisor state except in a controlled manner, the instructions which modify the whole state register are privileged. To aid in debugging programs which are to be used as operating systems, the move to user stack pointer (MOVE to USP) and move from user stack pointer (MOVE from USP) instructions are also privileged.

The bus cycles generated by an instruction executed in the user state are classified as user state references. This allows an external memory management device to translate the address and to control access to protected portions of the address space. While the processor is in the user privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly, access the user stack pointer.

5.1.3 Privilege State Changes

Once the processor is in the user state and executing instructions, only exception processing can change the privilege state. During exception processing, the current setting of the S bit of the status register is saved and the S bit is asserted, putting the processor in the supervisor state. Therefore, when instruction execution resumes at the address specified to process the exception, the processor is in the supervisor privilege state.

5.1.4 Reference Classification

When the processor makes a reference, it classifies the kind of reference being made, using the encoding on the three function code output lines. This allows external translation of addresses, control of access, and differentiation of special processor state, such as interrupt acknowledge. Table 5-1 lists the classification of references.

Table 5-1. Bus Cycle Classification

Function Code Output			Reference Class
FC2	FC1	FC0	
0	0	0	(Unassigned)
0	0	1	User Data
0	1	0	User Program
0	1	1	(Unassigned)

Function Code Output			Reference Class
FC2	FC1	FC0	
1	0	0	(Unassigned)
1	0	1	Supervisor Data
1	1	0	Supervisor Program
1	1	1	Interrupt Acknowledge

5.2 EXCEPTION PROCESSING

Before discussing the details of interrupts, traps, and tracing, a general description of exception processing is in order. The processing of an exception occurs in four steps, with variations for different exception causes. During the first step, a temporary copy of the status register is made and the status register is set for exception processing. In the second step the exception vector is determined and the third step is the saving of the current processor context. In the fourth step a new context is obtained and the processor switches to instruction processing.

5.2.1 Exception Vectors

Exception vectors are memory locations from which the processor fetches the address of a routine which will handle that exception. All exception vectors are two words in length (Figure 5-1), except for the reset vector which is four words. All exception vectors lie in the supervisor data space, except for the reset vector which is in the supervisor program space. A vector number is an 8-bit number which, when multiplied by four, gives the address of an exception vector. Vector numbers are generated internally or externally, depending on the cause of the exception. In the case of interrupts, during the interrupt acknowledge bus cycle, a peripheral provides an 8-bit vector number (Figure 5-2) to the processor on data bus lines D0 through D7. The processor translates the vector number into a full 24-bit address, shown in Figure 5-3. The memory layout for exception vectors is given in Table 5-2.

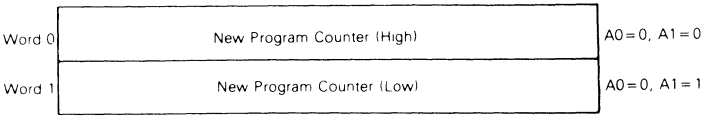


Figure 5-1. Format of Vector Table Entries

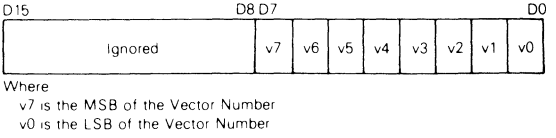


Figure 5-2. Vector Number Format

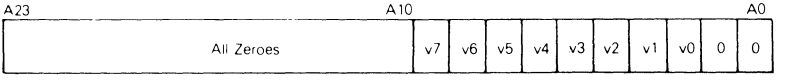


Figure 5-3. Exception Vector Address Calculation

As shown in Table 5-2, the memory layout is 512 words long (1024 bytes). It starts at address 0 and proceeds through address 1023. This provides 255 unique vectors; some of these are reserved for TRAPS and other system functions. Of the 255, there are 192 reserved for user interrupt vectors. However, there is no protection on the first 64 entries, so user interrupt vectors may overlap at the discretion of the systems designer.

5.2.2 Kinds of Exceptions

Exceptions can be generated by either internal or external causes. The externally generated exceptions are the interrupts and the bus error and reset requests. The interrupts are requests from

Table 5-2. Exception Vector Table

Vector Number(s)	Address			Assignment
	Dec	Hex	Space	
0	0	000	SP	Reset: Initial SSP
–	4	004	SP	Reset: Initial PC
2	8	008	SD	Bus Error
3	12	00C	SD	Address Error
4	16	010	SD	Illegal Instruction
5	20	014	SD	Zero Divide
6	24	018	SD	CHK Instruction
7	28	01C	SD	TRAPV Instruction
8	32	020	SD	Privilege Violation
9	36	024	SD	Trace
10	40	028	SD	Line 1010 Emulator
11	44	02C	SD	Line 1111 Emulator
12*	48	030	SD	(Unassigned, Reserved)
13*	52	034	SD	(Unassigned, Reserved)
14*	56	038	SD	(Unassigned, Reserved)
15	60	03C	SD	Uninitialized Interrupt Vector
16-23*	64	04C	SD	(Unassigned, Reserved)
	95	05F		–
24	96	060	SD	Spurious Interrupt
25	100	064	SD	Level 1 Interrupt Autovector
26	104	068	SD	Level 2 Interrupt Autovector
27	108	06C	SD	Level 3 Interrupt Autovector
28	112	070	SD	Level 4 Interrupt Autovector
29	116	074	SD	Level 5 Interrupt Autovector
30	120	078	SD	Level 6 Interrupt Autovector
31	124	07C	SD	Level 7 Interrupt Autovector
32-47	128	080	SD	TRAP Instruction Vectors
	191	0BF		–
48-63*	192	0C0	SD	(Unassigned, Reserved)
	255	0FF		–
64-255	256	100	SD	User Interrupt Vectors
	1023	3FF		–

*Vector numbers 12 13 14 16 through 23, and 48 through 63 are reserved for future enhancements by THOMSON SEMICONDUCTEURS. No user peripheral devices should be assigned these numbers.

peripheral devices for processor action while the bus error and reset inputs are used for access control and processor restart. The internally generated exceptions come from instructions, or from address errors or tracing. The trap (TRAP), trap on overflow (TRAPV), check data register against upper bounds (CHK), and divide (DIV) instructions all can generate exceptions as part of their instruction execution. In addition, illegal instructions, word fetches from odd addresses, and privilege violations cause exceptions. Tracing behaves like a very high-priority internally-generated interrupt after each instruction execution.

5.2.3 Exception Processing Sequence

Exception processing occurs in four identifiable steps. In the first step, an internal copy is made of the status register. After the copy is made, the S bit is asserted, putting the processor into the supervisor privilege state. Also, the T bit is negated which will allow the exception handler to execute unhindered by tracing. For the reset and interrupt exceptions, the interrupt priority mask is also updated.

In the second step, the vector number of the exception is determined. For interrupts, the vector number is obtained by a processor fetch and classified as an interrupt acknowledge. For all other exceptions, internal logic provides the vector number. This vector number is then used to generate the address of the exception vector.

The third step is to save the current processor status, except for the reset exception. The current program counter value and the saved copy of the status register are stacked using the supervisor stack pointer as shown in Figure 5-4. The program counter value stacked usually points to the next unexecuted instruction; however, for bus error and address error, the value stacked for the program counter is unpredictable, and may be incremented from the address of the instruction which caused the error. Additional information defining the current context is stacked for the bus error and address error exceptions.

The last step is the same for all exceptions. The new program counter value is fetched from the exception vector. The processor then resumes instruction execution. The instruction at the address given in the exception vector is fetched, and normal instruction decoding and execution is started.

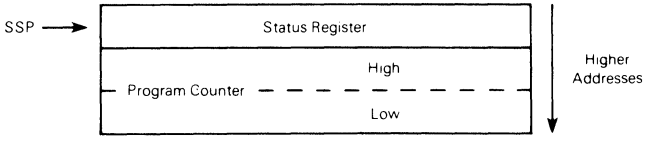


Figure 5-4. Exception Stack Order (Groups 1 and 2)

5.2.4 Multiple Exceptions

These paragraphs describe the processing which occurs when multiple exceptions arise simultaneously. Exceptions can be grouped according to their occurrence and priority. The group 0 exceptions are reset, bus error, and address error. These exceptions cause the instruction currently being executed to be aborted and the exception processing to commence within two clock cycles.

The group 1 exceptions are trace and interrupt, as well as the privilege violations and illegal instructions. These exceptions allow the current instruction to execute to completion, but pre-empt the execution of the next instruction by forcing exception processing to occur (privilege violations and illegal instructions are detected when they are the next instruction to be executed). The group 2 exceptions occur as part of the normal processing of instructions. The TRAP, TRAPV, CHK, and zero divide exceptions are in this group. For these exceptions, the normal execution of an instruction may lead to exception processing.

Group 0 exceptions have highest priority, while group 2 exceptions have lowest priority. Within group 0, reset has highest priority, followed by bus error and then address error. Within group 1, trace has priority over external interrupts, which in turn takes priority over illegal instruction and privilege violation. Since only one instruction can be executed at a time, there is no priority relation within group 2.

The priority relation between two exceptions determines which is taken, or taken first, if the conditions for both arise simultaneously. Therefore, if a bus error occurs during a TRAP instruction, the bus error takes precedence, and the TRAP instruction processing is aborted. In another example, if an interrupt request occurs during the execution of an instruction while the T bit is asserted, the trace exception has priority, and is processed first. Before instruction processing resumes, however, the interrupt exception is also processed, and instruction processing commences finally in the interrupt handler routine. A summary of exception grouping and priority is given in Table 5-3.

Table 5-3. Exception Grouping and Priority

Group	Exception	Processing
0	Reset Address Error Bus Error	Exception processing begins within two clock cycles
1	Trace Interrupt Illegal Privilege	Exception processing begins before the next instruction
2	TRAP, TRAPV, CHK, Zero Divide	Exception processing is started by normal instruction execution

5.3 EXCEPTION PROCESSING DETAILED DISCUSSION

Exceptions have a number of sources and each exception has processing which is peculiar to it. The following paragraphs detail the sources of exceptions, how each arises, and how each is processed.

5.3.1 Reset

The reset input provides the highest exception level. The processing of the reset signal is designed for system initiation and recovery from catastrophic failure. Any processing in progress at the time of the reset is aborted and cannot be recovered. The processor is forced into the supervisor state and the trace state is forced off. The processor interrupt priority mask is set at level seven. The vector number is internally generated to reference the reset exception vector at location 0 in the supervisor program space. Because no assumptions can be made about the validity of register contents,

in particular the supervisor stack pointer, neither the program counter nor the status register is saved. The address contained in the first two words of the reset exception vector is fetched as the initial supervisor stack pointer, and the address in the last two words of the reset exception vector is fetched as the initial program counter. Finally, instruction execution is started at the address in the program counter. The power-up/restart code should be pointed to by the initial program counter.

The reset instruction does not cause loading of the reset vector, but does assert the reset line to reset external devices. This allows the software to reset the system to a known state and then continue processing with the next instruction.

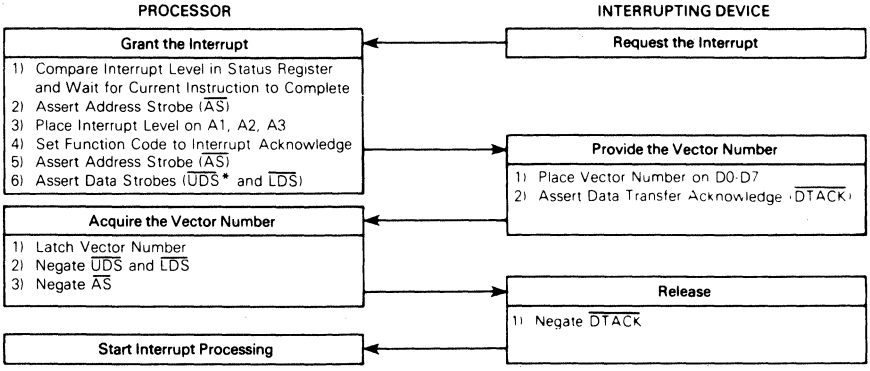
5.3.2 Interrupts

Seven levels of interrupt priorities are provided. Devices may be chained externally within interrupt priority levels, allowing an unlimited number of peripheral devices to interrupt the processor. Interrupt priority levels are numbered from one to seven, with level seven being the highest priority. The status register contains a 3-bit mask which indicates the current processor priority, and interrupts are inhibited for all priority levels less than or equal to the current processor priority.

An interrupt request is made to the processor by encoding the interrupt request level on the interrupt request lines; a zero indicates no interrupt request. Interrupt requests arriving at the processor do not force immediate exception processing, but are made pending. Pending interrupts are detected between instruction executions. If the priority of the pending interrupt is lower than or equal to the current processor priority, execution continues with the next instruction and the interrupt exception processing is postponed. (The recognition of level seven is slightly different, as explained in the following paragraph.)

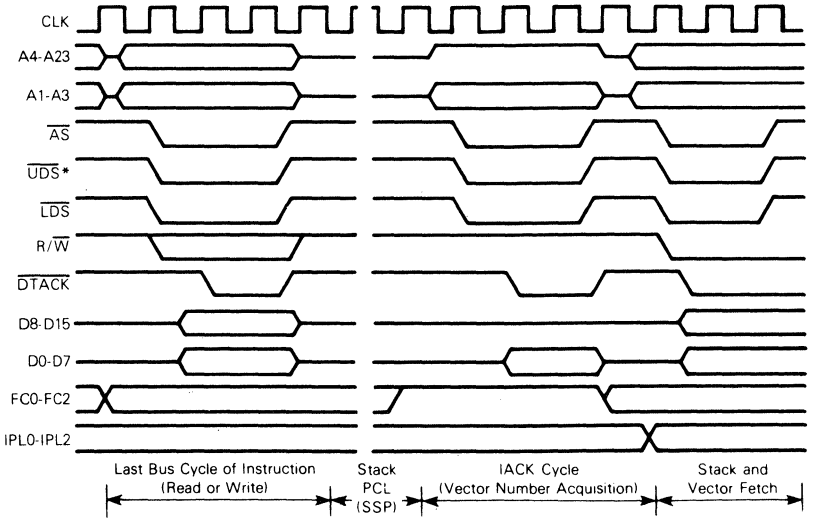
If the priority of the pending interrupt is greater than the current processor priority, the exception processing sequence is started. A copy of the status register is saved, the privilege state is sent to the supervisor stack, tracing is suppressed, and the processor priority level is set to the level of the interrupt acknowledged. The processor fetches the vector number from the interrupting device, classifying the reference as an interrupt acknowledge and displaying the level number of the interrupt being acknowledged on the address bus. If external logic requests an automatic vectoring, the processor internally generates a vector number which is determined by the interrupt level number. If external logic indicates a bus error, the interrupt is taken to be spurious, and the generated vector number references the spurious interrupt vector. The processor then proceeds with the usual exception processing, saving the program counter and status register on the supervisor stack. The saved value of the program counter is the address of the instruction which would have been executed had the interrupt not been present. The content of the interrupt vector whose vector number was previously obtained is fetched and loaded into the program counter, and normal instruction execution commences in the interrupt handling routine. A flowchart for the interrupt acknowledge sequence is given in Figure 5-5, a timing diagram is given in Figure 5-6, and the interrupt processing sequence is shown in Figure 5-7.

Priority level seven is a special case. Level seven interrupts cannot be inhibited by the interrupt priority mask, thus providing a "non-maskable interrupt" capability. An interrupt is generated each time the interrupt request level changes from some lower level to level seven. Note that a level seven interrupt may still be caused by the level comparison if the request level is a seven and the processor priority is set to a lower level by an instruction.



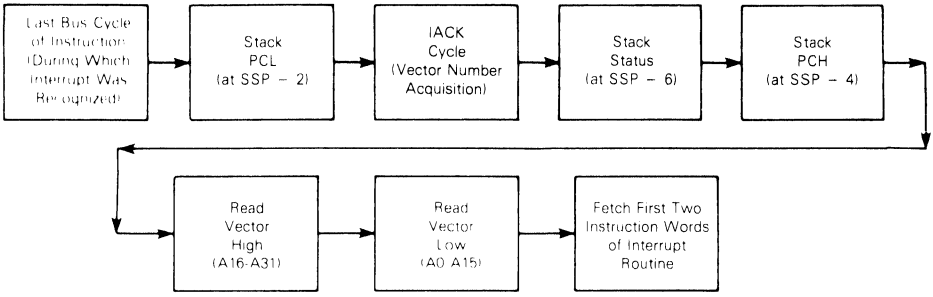
* Although a vector number is one byte, both data strobes are asserted due to the microcode used for exception processing. The processor does not recognize anything on data lines D8 through D15 at this time.

Figure 5-5. Vector Acquisition Flowchart



* Although a vector number is one byte, both data strobes are asserted due to the microcode used for exception processing. The processor does not recognize anything on data lines D8 through D15 at this time.

Figure 5-6. Interrupt Acknowledge Cycle Timing Diagram



NOTE: SSP refers to the value of the supervisor stack pointer before the interrupt occurs.

Figure 5-7. Interrupt Processing Sequence

5.3.3 Uninitialized Interrupt

An interrupting device asserts $\overline{VPA}$ or provides an interrupt during an interrupt acknowledge cycle to the TS68000. If the vector register has not been initialized, the responding TS68000 Family peripheral will provide vector 15, the uninitialized interrupt vector. This provides a uniform way to recover from a programming error.

5.3.4 Spurious Interrupt

If during the interrupt acknowledge cycle no device responds by asserting $\overline{DTACK}$ or $\overline{VPA}$, the bus error line should be asserted to terminate the vector acquisition. The processor separates the processing of this error from bus error by fetching the spurious interrupt vector instead of the bus error vector. The processor then proceeds with the usual exception processing.

5.3.5 Instruction Traps

Traps are exceptions caused by instructions. They arise either from processor recognition of abnormal conditions during instruction execution, or from use of instructions whose normal behavior is trapping.

Some instructions are used specifically to generate traps. The TRAP instruction always forces an exception and is useful for implementing system calls for user programs. The TRAPV and CHK instructions force an exception if the user program detects a runtime error, which may be an arithemetic overflow or a subscript out of bounds.

The signed divide (DIVS) and unsigned (DIVU) instructions will force an exception if a division operation is attempted with a divisor of zero.

5.3.6 Illegal and Unimplemented Instructions

"Illegal instruction" is the term used to refer to any of the word bit patterns which are not the bit pattern of the first word of a legal instruction. During instruction execution, if such an instruction is

fetched, an illegal instruction exception occurs. THOMSON SEMICONDUCTEURS reserves the right to define instructions whose opcodes may be any of the illegal instructions. Three bit patterns will always force an illegal instruction trap on all TS68000 Family compatible microprocessors. They are : $4AFA, $4AFB, and $4AFC. Two of the patterns, $4AFA and $4AFB, are reserved for THOMSON SEMICONDUCTEURS system products. The third pattern, S4AFC, is reserved for customer use.

Word patterns with bits 15 through 12 equaling 1010 or 1111 are distinguished as unimplemented instructions and separate exception vectors are given to these patterns to permit efficient emulation. This facility allows the operating system to detect program errors, or to emulate unimplemented instructions in software.

5.3.7 Privilege Violations

In order to provide system security, various instructions are privileged. An attempt to execute one of the privileged instructions while in the user state will cause an exception. The privileged instructions are:

STOP	AND Immediate to SR
RESET	EOR Immediate to SR
RTE	OR Immediate to SR
MOVE to SR	MOVE USP

5.3.8 Tracing

To aid in program development, the TS68000 includes a facility to allow instruction-by-instruction tracing. In the trace state, after each instruction is executed an exception is forced, allowing a debugging program to monitor the execution of the program under test.

The trace facility uses the T bit in the supervisor portion of the status register. If the T bit is negated (off), tracing is disabled, and instruction execution proceeds from instruction to instruction as normal. If the T bit is asserted (on) at the beginning of the execution of an instruction, a trace exception will be generated after the execution of that instruction is completed. If the instruction is not executed, either because an interrupt is taken, or the instruction is illegal or privileged, the trace exception does not occur. The trace exception also does not occur if the instruction is aborted by a reset, bus error, or address error exception. If the instruction is indeed executed and an interrupt is pending on completion, the trace exception is processed before the interrupt exception. If, during the execution of the instruction an exception is forced by that instruction, the forced exception is processed before the trace exception.

As an extreme illustration of the above rules, consider the arrival of an interrupt during the execution of a TRAP instruction while tracing is enabled. First the trap exception is processed, then the trace exception, and finally the interrupt exception. Instruction execution resumes in the interrupt handler routine.

5.3.9 Bus Error

Bus error exceptions occur when the external logic requests that a bus error be processed by an exception. The current bus cycle which the processor is making is then aborted. Whether the processor was doing instruction or exception processing, that processing is terminated, and the processor immediately begins exception processing.

Exception processing for the bus error follows the usual sequence of steps. The status register is copied, the supervisor state is entered, and the trace state is turned off. The vector number is generated to refer to the bus error vector. Since the processor was not between instructions when the bus error exception request was made, the context of the processor is more detailed. To save more of this context, additional information is saved on the supervisor stack. The program counter and the copy of the status register are of course saved. The value saved for the program counter is advanced by some amount, one to five words beyond the address of the first word of the instruction which made the reference causing the bus error. If the bus error occurred during the fetch of the next instruction, the saved program counter has a value in the vicinity of the current instruction, even if the current instruction is a branch, a jump, or a return instruction. Besides the usual information, the processor saves its internal copy of the first word of the instruction being processed and the address which was being accessed by the aborted bus cycle. Specific information about the access is also saved: whether it was a read or a write, whether or not the processor was processing an instruction, and the classification displayed on the function code outputs when the bus error occurred. The processor is processing an instruction if it is in the normal state or processing a group 2 exception; the processor is not processing an instruction if it is processing a group 0 or a group 1 exception. Figure 5-8 illustrates how this information is organized on the supervisor stack. Although this information is not sufficient in general to effect full recovery from the bus error, it does allow software diagnosis. Finally, the processor commences instruction processing at the address contained in vector number two. It is the responsibility of the error handler routine to clean up the stack and determine where to continue execution.

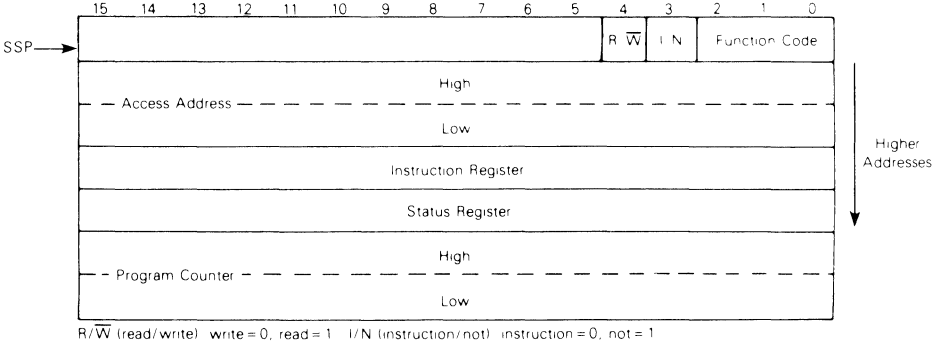


Figure 5-8. Exception Stack Order (Group 0)

If a bus error occurs during the exception processing for a bus error, address error, or reset, the processor is halted and all processing ceases. This simplifies the detection of catastrophic system failure, since the processor removes itself from the system rather than destroy any memory contents. Only the RESET pin can restart a halted processor.

5.3.10 Address Error

Address error exceptions occur when the processor attempts to access a word or a long word operand or an instruction at an odd address. The effect is much like an internally generated bus

error, so that the bus cycle is aborted and the processor ceases whatever processing it is currently doing and begins exception processing. After the exception processing commences, the sequence is the same as that for bus error including the information that is stacked, except that the vector number refers to the address error vector instead. Likewise, if an address error occurs during the exception processing for a bus error, address error, or reset, the processor is halted. As shown in Figure 5-9, an address error will execute a short bus cycle followed by exception processing.

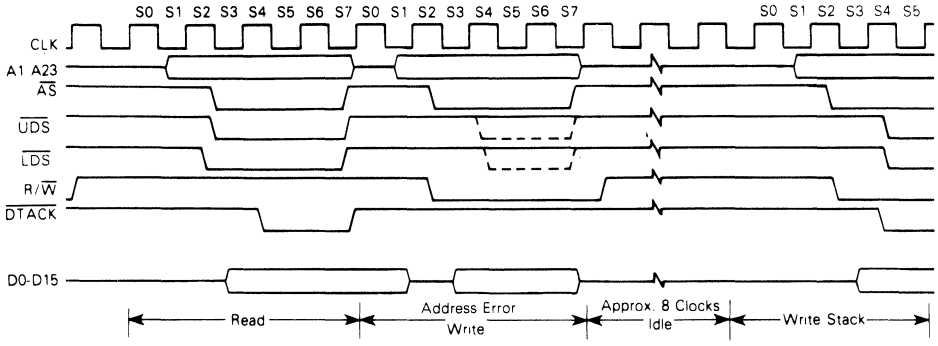


Figure 5-9. Address Error Timing Diagram

SECTION 6
INTERFACE WITH EF6800 PERIPHERALS

THOMSON SEMICONDUCTEURS' extensive line of EF6800 peripherals are directly compatible with the TS68000. Some of these devices that are particularly useful are :

EF6821 Peripheral Interface Adapter
EF6840 Programmable Timer Module
EF9345, EF9367 CRT Controllers
EF6850 Asynchronous Communications Interface Adapter
EF6852 Synchronous Serial Data Adapter
EF6854 Advanced Data Link Controller

To interface the synchronous EF6800 peripherals with the asynchronous TS68000, the processor modifies its bus cycle to meet the EF6800 cycle requirements whenever an EF6800 device address is detected. This is possible since both processors use memory mapped I/O. Figure 6-1 is a flowchart of the interface operation between the processor and EF6800 devices.

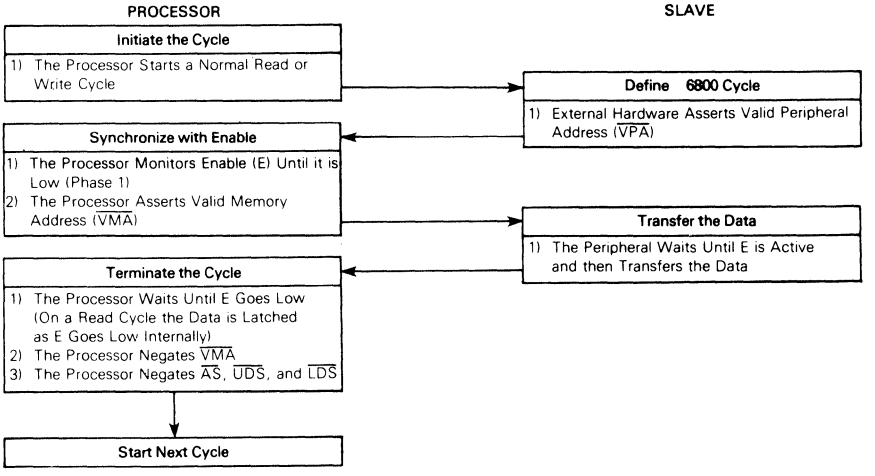


Figure 6-1. EF6800 Interfacing Flowchart

6.1 DATA TRANSFER OPERATION

Three signals on the processor provide the EF6800 interface. They are : enable (E), valid memory address ($\overline{VMA}$), and valid peripheral address ($\overline{VPA}$). Enable corresponds to the E or phase 2 signal in existing 6800 systems. The bus frequency in one tenth of the incoming TS68000 clock frequency. The timing of E allows 1 megahertz peripherals to be used with 8 megahertz TS68000s. Enable has a 60/40 duty cycle ; that is, it is low for six input clocks and high for four input clocks. This duty cycle allows the processor to do successive $\overline{VPA}$ accesses on successive E pulses.

EF6800 cycle timing is given in Figures 6-2, 6-3, 8-7, and 8-8. At state zero (S0) in the cycle, the address bus is in the high-impedence state. A function code is asserted on the function code output lines. One-half clock later, in state 1, the address bus is released from the high-impedence state.

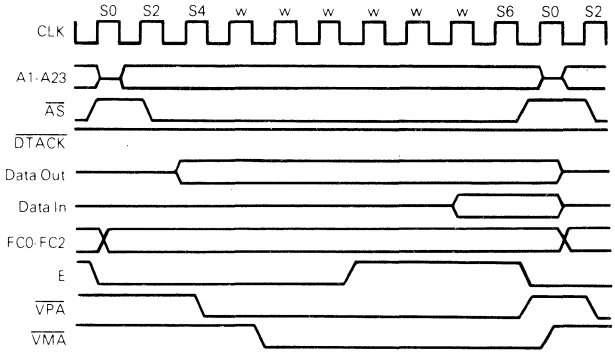


Figure 6-2. TS68000 to EF6800 Peripheral Timing — Best Case

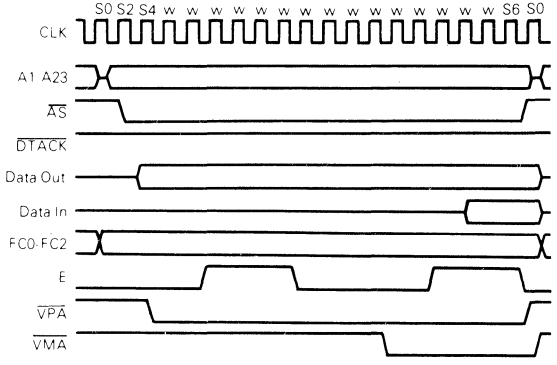


Figure 6-3. TS68000 to EF6800 Peripheral Timing — Worst Case

During state 2, the address strobe ($\overline{AS}$) is asserted to indicate that there is a valid address on the address bus. If the bus cycle is a read cycle, the upper and/or lower data strobes are also asserted in state 2. If the bus cycle is a write cycle, the read/write (R/$\overline{W}$) signal is switched to low (write) during state 2. One-half clock later, in state 3, the write data is placed on the data bus, and in state 4 the data strobes are issued to indicate valid data on the data bus. The processor now inserts wait states until it recognizes the assertion of $\overline{VPA}$.

The $\overline{VPA}$ input signals the processor that the address on the bus is the address of an EF6800 device (or an area reserved for EF6800 devices) and that the bus should conform to the phase 2 transfer characteristics of the EF6800 bus. Valid peripheral address is derived by decoding the address bus, conditioned by the address strobe. Chip select for the EF6800 peripherals should be derived by decoding the address bus conditioned by $\overline{VMA}$.

After recognition of $\overline{VPA}$, the processor assures that the enable (E) is low, by waiting if necessary, and subsequently asserts $\overline{VMA}$. Valid memory address is then used as part of the chip select equation of the peripheral. This ensures that the EF6800 peripherals are selected and deselected at the correct time. The peripheral now runs its cycle during the high portion of the E signal. Figures 6-2 and 6-3 depict the best and worst case EF6800 cycle timing. This cycle length is dependent strictly upon when $\overline{VPA}$ is asserted in relationship to the E clock.

If we assume that external circuitry asserts $\overline{VPA}$ as soon as possible after the assertion of $\overline{AS}$, then $\overline{VPA}$ will be recognized as being asserted on the falling edge of S4. In this case, no "extra" wait cycles will be inserted prior to the recognition of $\overline{VPA}$ asserted and only the wait cycles inserted to synchronize with the E clock will determine the total length of the cycle. In any case, the synchronization delay will be some integral number of clock cycles within the following two extremes:

1. Best Case — $\overline{VPA}$ is recognized as being asserted on the falling edge three clock cycles before E rises (or three clock cycles after E falls).
2. Worst Case — $\overline{VPA}$ is recognized as being asserted on the falling edge two clock cycles before E rises (or four clock cycles after E falls).

During a read cycle, the processor latches the peripheral data in state 6. For all cycles, the processor negates the address and data strobes one-half clock cycle later in state 7 and the enable signal goes low at this time. Another half clock later, the address bus is put in the high-impedence state. During a write cycle, the data bus is put in the high-impedence state and the read/write signal is switched high. The peripheral logic must remove $\overline{VPA}$ within one clock after the address strobe is negated.

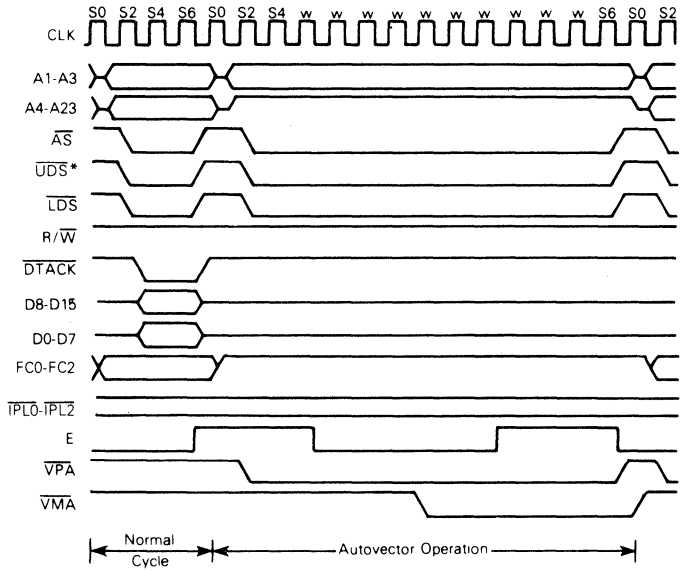
$\overline{DTACK}$ should not be asserted while $\overline{VPA}$ is asserted. Notice that the TS68000 $\overline{VMA}$ is active low, contrasted with the active high EF6800 $\overline{VMA}$. This allows the processor to put its buses in the high-impedence state on DMA requests without inadvertently selecting the peripherals.

6.2 INTERRUPT INTERFACE OPERATION

During an interrupt acknowledge cycle while the processor is fetching the vector, the $\overline{VPA}$ is asserted, the TS68000 will assert $\overline{VMA}$ and complete a normal EF6800 read cycle as shown in Figure 6-4. The processor will then use an internally generated vector that is a function of the interrupt being serviced. This process is known as autovectoring. The seven autovectors are vector numbers 25 through 31 (decimal).

Autovectoring operates in the same fashion (but is not restricted to) the EF6800 interrupt sequence The basic difference is that there are six normalinterrupt vectors and one NMI type vector. As with both the EF6800 and the TS68000's normal vectored interrupt, the interrupt service routine can be located anywhere in the address space. This is due to the fact that while the vector numbers are fixed, the contents of the vector table entries are assigned by the user.

Since $\overline{VMA}$ is asserted during autovectoring, care should be taken to insure that the 6800 peripheral address decoding prevents unintended accesses.



* Although $\overline{UDS}$ and $\overline{LDS}$ are asserted, no data is read from the bus during the autovector cycle The vector number is generated internally.

Figure 6-4. Autovector Operation Timing Diagram

SECTION 7
INSTRUCTION SET AND EXECUTION TIMES

7.1 INSTRUCTION SET

The following paragraphs provide information about the addressing categories and instruction set of the TS68000.

7.1.1 Addressing Categories

Effective address modes may be categorized by the ways in which they may be used. The following classifications will be used in the instruction definitions.

Data — If an effective address mode may be used to refer to data operands, it is considered a data addressing effective address mode.

Memory — If an effective address mode may be used to refer to memory operands, it is considered a memory addressing effective address mode.

Alterable — If an effective address mode may be used to refer to alterable (writeable) operands, it is considered an alterable addressing effective address mode.

Control — If an effective address mode may be used to refer to memory operands without an associated size, it is considered a control addressing effective address mode.

These categories may be combined, so that additional, more restrictive, classifications may be defined. For example, the instruction descriptions use such classifications as alterable memory or data alterable. The former refers to those addressing modes which are both alterable and memory addresses, and the latter refers to addressing modes which are both data and alterable.

Table 7-1 shows the various categories to which each of the effective address modes belong. Table 7-2 is the instruction set summary.

Table 7-1. Effective Addressing Mode Categories

Effective Address Modes	Mode	Register	Addressing Categories			
			Data	Memory	Control	Alterable
Dn	000	Register Number	X	–	–	X
An	001	Register Number	–	–	–	X
(An)	010	Register Number	X	X	X	X
(An) +	011	Register Number	X	X	–	X
– (An)	100	Register Number	X	X	–	X
d(An)	101	Register Number	X	X	X	X
d(An, ix)	110	Register Number	X	X	X	X
xxx.W	111	000	X	X	X	X
xxx.L	111	001	X	X	X	X
d(PC)	111	010	X	X	X	–
d(PC, ix)	111	011	X	X	X	–
#xxx	111	X	X	X	–	–

Table 7-2. Instruction Set (Sheet 1 of 2)

Mnemonic	Description	Operation	Condition Codes				
			X	N	Z	V	C
ABCD	Add Decimal with Extend	(Destination)$_{10}$ + (Source)$_{10}$ + X → Destination	*	U	*	U	*
ADD	Add Binary	(Destination) + (Source) → Destination	*	*	*	*	*
ADDA	Add Address	(Destination) + (Source) → Destination	–	–	–	–	–
ADDI	Add Immediate	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDQ	Add Quick	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDX	Add Extended	(Destination) + (Source) + X → Destination	*	*	*	*	*
AND	AND Logical	(Destination) Λ (Source) → Destination	–	*	*	0	0
ANDI	AND Immediate	(Destination) Λ Immediate Data → Destination	–	*	*	0	0
ANDI to CCR	AND Immediate to Condition Codes	(Source) Λ CCR → CCR	*	*	*	*	*
ANDI to SR	AND Immediate to Status Register	(Source) Λ SR → SR	*	*	*	*	*
ASL, ASR	Arithmetic Shift	(Destination) Shifted by <count> → Destination	*	*	*	*	*
B$_{CC}$	Branch Conditionally	If $_{CC}$ then PC + d → PC	–	–	–	–	–
BCHG	Test a Bit and Change	~(<bit number>) OF Destination → Z ~(<bit number>) OF Destination → <bit number> OF Destination	–	–	*	–	–
BCLR	Test a Bit and Clear	~(<bit number>) OF Destination → Z 0 → <bit number> → OF Destination	–	–	*	–	–
BRA	Branch Always	PC + d → PC	–	–	–	–	–
BSET	Test a Bit and Set	~(<bit number>) OF Destination → Z 1 → <bit number> OF Destination	–	–	*	–	–
BSR	Branch to Subroutine	PC → – (SP); PC + d → PC	–	–	–	–	–
BTST	Test a Bit	~(<bit number>) OF Destination → Z	–	–	*	–	–
CHK	Check Register Against Bounds	If Dn <0 or Dn> (<ea>) then TRAP	–	*	U	U	U
CLR	Clear and Operand	0 → Destination	–	0	1	0	0
CMP	Compare	(Destination) – (Source)	–	*	*	*	*
CMPA	Compare Address	(Destination) – (Source)	–	*	*	*	*
CMPI	Compare Immediate	(Destination) – Immediate Data	–	*	*	*	*
CMPM	Compare Memory	(Destination) – (Source)	–	*	*	*	*
DB$_{CC}$	Test Condition, Decrement and Branch	If ~$_{CC}$ then Dn – 1 → Dn; if Dn ≠ – 1 then PC + d → PC	–	–	–	–	–
DIVS	Signed Divide	(Destination)/(Source) → Destination	–	*	*	*	0
DIVU	Unsigned Divide	(Destination)/(Source) → Destination	–	*	*	*	0
EOR	Exclusive OR Logical	(Destination) ⊕ (Source) → Destination	–	*	*	0	0
EORI	Exclusive OR Immediate	(Destination) ⊕ Immediate Data → Destination	–	*	*	0	0
EORI to CCR	Exclusive OR Immediate to Condition Codes	(Source) ⊕ CCR → CCR	*	*	*	*	*
EORI to SR	Exclusive OR Immediate to Status Register	(Source) ⊕ SR → SR	*	*	*	*	*
EXG	Exchange Register	Rx ↔ Ry	–	–	–	–	–
EXT	Sign Extend	(Destination) Sign-Extended → Destination	–	*	*	0	0
JMP	Jump	Destination → PC	–	–	–	–	–
JSR	Jump to Subroutine	PC → – (SP); Destination → PC	–	–	–	–	–
LEA	Load Effective Address	<ea> → An	–	–	–	–	–
LINK	Link and Allocate	An → – (SP); SP → An; SP + Displacement → SP	–	–	–	–	–
LSL, LSR	Logical Shift	(Destination) Shifted by <count> → Destination	*	*	*	0	*
MOVE	Move Data from Source to Destination	(Source) → Destination	–	*	*	0	0
MOVE to CCR	Move to Condition Code	(Source) → CCR	*	*	*	*	*
MOVE to SR	Move to the Status Register	(Source) → SR	*	*	*	*	*

Λ logical AND
V logical OR
⊕ logical exclusive OR
~ logical complement

* affected
– unaffected
0 cleared
1 set
U undefined

Table 7-2. Instruction Set (Sheet 2 of 2)

Mnemonic	Description	Operation	Condition Codes				
			X	N	Z	V	C
MOVE from SR	Move from the Status Register	SR → Destination	-	-	-	-	-
MOVE USP	Move User Stack Pointer	USP → An, An → USP	-	-	-	-	-
MOVEA	Move Address	(Source) → Destination	-	-	-	-	-
MOVEM	Move Multiple Registers	Registers → Destination (Source) → Registers	-	-	-	-	
MOVEP	Move Peripheral Data	(Source) → Destination	-	-	-	-	-
MOVEQ	Move Quick	Immediate Data → Destination	-	*	*	0	0
MULS	Signed Multiply	(Destination)X(Source) → Destination	-	*	*	0	0
MULU	Unsigned Multiply	(Destination)X(Source) → Destination	-	*	*	0	0
NBCD	Negate Decimal with Extend	$0 - (\text{Destination})_{10} - X \rightarrow$ Destination	*	U	*	U	*
NEG	Negate	0 – (Destination) → Destination	*	*	*	*	*
NEGX	Negate with Extend	0 – (Destination) – X → Destination	*	*	*	*	*
NOP	No Operation	–	-	-	-	-	-
NOT	Logical Complement	~ (Destination) → Destination	-	*	*	0	0
OR	Inclusive OR Logical	(Destination) v (Source) → Destination	-	*	*	0	0
ORI	Inclusive OR Immediate	(Destination) v Immediate Data → Destination	-	*	*	0	0
ORI to CCR	Inclusive OR Immediate to Condition Codes	(Source) v CCR → CCR	*	*	*	*	*
ORI to SR	Inclusive OR Immediate to Status Register	(Source) v SR → SR	*	*	*	*	*
PEA	Push Effective Address	<ea> → – (SP)	-	-	-	-	-
RESET	Reset External Device	–	-	-	-	-	-
ROL, ROR	Rotate (Without Extend)	(Destination) Rotated by <count> → Destination	-	*	*	0	*
ROXL, ROXR	Rotate with Extend	(Destination) Rotated by <count> → Destination	*	*	*	0	*
RTE	Return from Exception	(SP) + → SR; (SP) + → PC	*	*	*	*	*
RTR	Return and Restore Condition Codes	(SP) + → CC; (SP) + → PC	*	*	*	*	*
RTS	Return from Subroutine	(SP) + → PC	-	-	-	-	-
SBCD	Subtract Decimal with Extend	$(\text{Destination})_{10} - (\text{Source})_{10} - X \rightarrow$ Destination	*	U	*	U	*
Scc	Set According to Condition	If cc then 1's → Destination else 0's → Destination	-	-	-	-	-
STOP	Load Status Register and Stop	Immediate Data → SR; STOP	*	*	*	*	*
SUB	Subtract Binary	(Destination) – (Source) → Destination	*	*	*	*	*
SUBA	Subtract Address	(Destination) – (Source) → Destination	-	-	-	-	-
SUBI	Subtract Immediate	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBQ	Subtract Quick	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBX	Subtract with Extend	(Destination) – (Source) – X → Destination	*	*	*	*	*
SWAP	Swap Register Halves	Register [31:16] ↔ Register [15:0]	-	*	*	0	0
TAS	Test and Set an Operand	(Destination) Tested → CC, 1 → [7] OF Destination	-	*	*	0	0
TRAP	Trap	PC → – (SSP); SR → – (SSP); (Vector) → PC	-	-	-	-	-
TRAPV	Trap on Overflow	If V then TRAP	-	-	-	-	-
TST	Test and Operand	(Destination) Tested → CC	-	*	*	0	0
UNLK	Unlink	An → SP; (SP) + → An	-	-	-	-	-

[ ] = bit number
Λ logical AND
V logical OR
⊕ logical exclusive OR
~ logical complement

* affected
– unaffected
0 cleared
1 set
U undefined

7.1.2 Instruction Prefetch

The TS68000 uses a two-word tightly-coupled instruction prefetch mechanism to enhance performance. This mechanism is described in terms of the microcode operations involved. If the execution of an instruction is defined to begin when the microroutine for that instruction is entered, some features of the prefetch mechanism can be described.

1. When execution of an instruction begins, the operation word and the word following have already been fetched. The operation word is in the instruction decoder.
2. In the case of multi-word instructions, as each additional word of the instruction is used internally, a fetch is made to the instruction stream to replace it.
3. The last fetch for an instruction from the instruction stream is made when the operation word is discarded and decoding is started on the next instruction.
4. If the instruction is a single-word instruction causing a branch, the second word is not used. But because this word is fetched by the preceding instruction, it is impossible to avoid this superfluous fetch.
5. In the case of an interrupt or trace exception, both words are not used.
6. The program counter usually points to the last word fetched from the instruction stream.

7.2 INSTRUCTION EXECUTION TIMES

The following paragraphs contain listings of the instruction execution times in terms of external clock (CLK) periods. In this timing data, it is assumed that both memory read and write cycle times are four clock periods. Any wait states caused by a longer memory cycle must be added to the total instruction time. The number of bus read and write cycles for each instruction is also included with the timing data. This timing data is enclosed in parenthesis following the execution periods and is shown as (r/w) where r is the number of read cycles and w is the number of write cycles.

NOTE

The number of periods includes instruction fetch and all applicable operand fetches and stores.

7.2.1 Effective Address Operand Calculation Timing

Table 7-3 lists the number of clock periods required to compute an instruction's effective address. It includes fetching of any extension words, the address computation, and fetching of the memory operand. The number of bus read and write cycles is shown in parenthesis as (r/w). Note there are no write cycles involved in processing the effective address.

7.2.2 Move Instruction Execution Times

Tables 7-4 and 7-5 indicate the number of clock periods for the move instruction. This data includes instruction fetch, operand reads, and operand writes. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 7-3. Effective Address Calculation Times

Addressing Mode		Byte, Word	Long
	Register		
Dn	Data Register Direct	0(0/0)	0(0/0)
An	Address Register Direct	0(0/0)	0(0/0)
	Memory		
(An)	Address Register Indirect	4(1/0)	8(2/0)
(An) +	Address Register Indirect with Postincrement	4(1/0)	8(2/0)
– (An)	Address Register Indirect with Predecrement	6(1/0)	10(2/0)
d(An)	Address Register Indirect with Displacement	8(2/0)	12(3/0)
d(An, ix)*	Address Register Indirect with Index	10(2/0)	14(3/0)
xxx W	Absolute Short	8(2/0)	12(3/0)
xxx L	Absolute Long	12(3/0)	16(4/0)
d(PC)	Program Counter with Displacement	8(2/0)	12(3/0)
d(PC, ix)*	Program Counter with Index	10(2/0)	14(3/0)
#xxx	Immediate	4(1/0)	8(2/0)

* The size of the index register (ix) does not affect execution time.

Table 7-4. Move Byte and Word Instruction Execution Times

Source	Destination								
	Dn	An	(An)	(An) +	– (An)	d(An)	d(An, ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
(An)	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
(An) +	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
– (An)	10(2/0)	10(2/0)	14(2/1)	14(2/1)	14(2/1)	18(3/1)	20(3/1)	18(3/1)	22(4/1)
d(An)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d(An, ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
xxx W	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
xxx L	16(4/0)	16(4/0)	20(4/1)	20(4/1)	20(4/1)	24(5/1)	26(5/1)	24(5/1)	28(6/1)
d(PC)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d(PC, ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
#xxx	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)

* The size of the index register (ix) does not affect execution time.

Table 7-5. Move Long Instruction Execution Times

Source	Destination								
	Dn	An	(An)	(An) +	– (An)	d(An)	d(An, ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	12(1/2)	12(1/2)	12(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An	4(1/0)	4(1/0)	12(1/2)	12(1/2)	12(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
(An)	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
(An) +	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
– (An)	14(3/0)	14(3/0)	22(3/2)	22(3/2)	22(3/2)	26(4/2)	28(4/2)	26(4/2)	30(5/2)
d(An)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
d(An, ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
xxx W	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
xxx L	20(5/0)	20(5/0)	28(5/2)	28(5/2)	28(5/2)	32(6/2)	34(6/2)	32(6/2)	36(7/2)
d(PC)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(5/2)
d(PC, ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
#xxx	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)

* The size of the index register (ix) does not affect execution time.

7.2.3 Standard Instruction Execution Times

The number of clock periods shown in Table 7-6 indicates the time required to perform the operations, store the results, and read the next instruction. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

In Table 7-6 the headings have the following meanings: An = address register operand, Dn = data register operand, ea = an operand specified by an effective address, and M = memory effective address operand.

Table 7-6. Standard Instruction Execution Times

Instruction	Size	op<ea>, An†	op<ea>, Dn	op Dn, <M>
ADD	Byte, Word	8(1/0)+	4(1/0)+	8(1/1)+
	Long	6(1/0)+**	6(1/0)+**	12(1/2)+
AND	Byte, Word	–	4(1/0)+	8(1/1)+
	Long	–	6(1/0)+**	12(1/2)+
CMP	Byte, Word	6(1/0)+	4(1/0)+	–
	Long	6(1/0)+	6(1/0)+	–
DIVS	–	–	158(1/0)+*	–
DIVU	–	–	140(1/0)+*	–
EOR	Byte, Word	–	4(1/0)***	8(1/1)+
	Long	–	8(1/0)***	12(1/2)+
MULS	–	–	70(1/0)+*	–
MULU	–	–	70(1/0)+*	–
OR	Byte, Word	–	4(1/0)+	8(1/1)+
	Long	–	6(1/0)+**	12(1/2)+
SUB	Byte, Word	8(1/0)+	4(1/0)+	8(1/1)+
	Long	6(1/0)+**	6(1/0)+**	12(1/2)+

NOTES:

+ add effective address calculation time
† word or long only
* indicates maximum value
** The base time of six clock periods is increased to eight if the effective address mode is register direct or immediate (effective address time should also be added).
*** Only available effective address mode is data register direct.
DIVS, DIVU – The divide algorithm used by the TS68000 provides less than 10 % difference between the best and worst case timings.
MULS, MULU – The multiply algorithm requires 38 + 2n clocks where n is defined as:
MULU: n = the number of ones in the <ea>
MULU: n = concatanate the <ea> with a zero as the LSB; n is the resultant number of 10 or 01 patterns in the 17-bit source; i.e., worst case happens when the source is $5555.

7.2.4 Immediate Instruction Execution Times

The number of clock periods shown in Table 7-7 includes the time to fetch immediate operands, perform the operations, store the results, and read the next operation. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

In Table 7-7, the headings have the following meanings: # = immediate operand, Dn = data register operand, An = address register operand, M = memory operand, and SR = status register.

Table 7-7. Immediate Instruction Execution Times

Instruction	Size	op #, Dn	op #, An	op #, M
ADDI	Byte, Word	8(2/0)	–	12(2/1)+
	Long	16(3/0)	–	20(3/2)+
ADDQ	Byte, Word	4(1/0)	8(1/0)*	8(1/1)+
	Long	8(1/0)	8(1/0)	12(1/2)+
ANDI	Byte, Word	8(2/0)	–	12(2/1)+
	Long	16(3/0)	–	20(3/1)+
CMPI	Byte, Word	8(2/0)	–	8(2/0)+
	Long	14(3/0)	–	12(3/0)+
EORI	Byte, Word	8(2/0)	–	12(2/1)+
	Long	16(3/0)	–	20(3/2)+
MOVEQ	Long	4(1/0)	–	–
ORI	Byte, Word	8(2/0)	–	12(2/1)+
	Long	16(3/0)	–	20(3/2)+
SUBI	Byte, Word	8(2/0)	–	12(2/1)+
	Long	16(3/0)	–	20(3/2)+
SUBQ	Byte, Word	4(1/0)	8(1/0)*	8(1/1)+
	Long	8(1/0)	8(1/0)	12(1/2)+

+ add effective address calculation time
*word only

7.2.5 Single Operand Instruction Execution Times

Table 7-8 indicates the number of clock periods for the single operand instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 7-8. Single Operand Instruction Execution Times

Instruction	Size	Register	Memory
CLR	Byte, Word	4(1/0)	8(1/1)+
	Long	6(1/0)	12(1/2)+
NBCD	Byte	6(1/0)	8(1/1)+
NEG	Byte, Word	4(1/0)	8(1/1)+
	Long	6(1/0)	12(1/2)+
NEGX	Byte, Word	4(1/0)	8(1/1)+
	Long	6(1/0)	12(1/2)+
NOT	Byte, Word	4(1/0)	8(1/1)+
	Long	6(1/0)	12(1/2)+
Scc	Byte, False	4(1/0)	8(1/1)+
	Byte, True	6(1/0)	8(1/1)+
TAS	Byte	4(1/0)	10(1/1)+
TST	Byte, Word	4(1/0)	4(1/0)+
	Long	4(1/0)	4(1/0)+

+ add effective address calculation time

7.2.6 Shift/Rotate Instruction Execution Times

Table 7-9 indicates the number of clock periods for the shift and rotate instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 7-9. Shift/Rotate Instruction Execution Times

Instruction	Size	Register	Memory
ASR, ASL	Byte, Word	6 + 2n(1/0)	8(1/1) +
	Long	8 + 2n(1/0)	–
LSR, LSL	Byte, Word	6 + 2n(1/0)	8(1/1) +
	Long	8 + 2n(1/0)	–
ROR, ROL	Byte, Word	6 + 2n(1/0)	8(1/1) +
	Long	8 + 2n(1/0)	–
ROXR, ROXL	Byte, Word	6 + 2n(1/0)	8(1/1) +
	Long	8 + 2n(1/0)	–

+ add effective address calculation time
n is the shift or rotate count

7.2.7 Bit Manipulation Instruction Execution Times

Table 7-10 indicates the number of clock periods required for the bit manipulation instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 7-10. Bit Manipulation Instruction Execution Times

Instruction	Size	Dynamic		Static	
		Register	Memory	Register	Memory
BCHG	Byte	–	8(1/1) +	–	12(2/1) +
	Long	8(1/0)*	–	12(2/0)*	–
BCLR	Byte	–	8(1/1) +	–	12(2/1) +
	Long	10(1/0)*	–	14(2/0)*	–
BSET	Byte	–	8(1/1) +	–	12(2/1) +
	Long	8(1/0)*	–	12(2/0)*	–
BTST	Byte	–	4(1/0) +	–	8(2/0) +
	Long	6(1/0)	–	10(2/0)	–

+ add effective address calculation time
* indicates maximum value

7.2.8 Conditional Instruction Execution Times

Table 7-11 indicates the number of clock periods required for the conditional instructions. The number of bus read and write cycles is indicated in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 7-11. Conditional Instruction Execution Times

Instruction	Displacement	Branch Taken	Branch Not Taken
B_{CC}	Byte	10(2/0)	8(1/0)
	Word	10(2/0)	12(2/0)
BRA	Byte	10(2/0)	–
	Word	10(2/0)	–
BSR	Byte	18(2/2)	–
	Word	18(2/2)	–
DB_{CC}	CC true	–	12(2/0)
	CC false	10(2/0)	14(3/0)

+ add effective address calculation time
* indicates maximum value

7.2.9 JMP, JSR, LEA, PEA, and MOVEM Instruction Execution Times

Table 7-12 indicates the number of clock periods required for the jump, jump-to-subroutine, load effective address, push effective address, and move multiple registers instructions. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 7-12. JMP, JSR, LEA, PEA, and MOVEM Instruction Execution Times

Instr	Size	(An)	(An)+	–(An)	d(An)	d(An, ix)+	xxx.W	xxx.L	d(PC)	d(PC, ix)*
JMP	–	8(2/0)	–	–	10(2/0)	14(3/0)	10(2/0)	12(3/0)	10(2/0)	14(3/0)
JSR	–	16(2/2)	–	–	18(2/2)	22(2/2)	18(2/2)	20(3/2)	18(2/2)	22(2/2)
LEA	–	4(1/0)	–	–	8(2/0)	12(2/0)	8(2/0)	12(3/0)	8(2/0)	12(2/0)
PEA	–	12(1/2)	–	–	16(2/2)	20(2/2)	16(2/2)	20(3/2)	16(2/2)	20(2/2)
MOVEM M→R	Word	12+4n (3+n/0)	12+4n (3+n/0)	–	16+4n (4+n/0)	18+4n (4+n/0)	16+4n (4+n/0)	20+4n (5+n/0)	16+4n (4+n/0)	18+4n (4+n/0)
	Long	12+8n (3+2n/0)	12+8n (3+2n/0)	–	16+8n (4+2n/0)	18+8n (4+2n/0)	16+8n (4+2n/0)	20+8n (5+2n/0)	16+8n (4+2n/0)	18+8n (4+2n/0)
MOVEM R→M	Word	8+4n (2/n)	–	8+4n (2/n)	12+4n (3/n)	14+4n (3/n)	12+4n (3/n)	16+4n (4/n)	– –	– –
	Long	8+8n (2/2n)	– –	8+8n (2/2n)	12+8n (3/2n)	14+8n (3/2n)	12+8n (3/2n)	16+8n (4/2n)	– –	– –

n is the number of registers to move
* is the size of the index register (ix) does not affect the instruction's execution time

7.2.10 Multi-Precision Instruction Execution Times

Table 7-13 indicates the number of clock periods for the multi-precision instructions. The number of clock periods includes the time to fetch both operands, perform the operations, store the results, and read the next instructions. The number of read and write cycles is shown in parenthesis as (r/w).

In Table 7-13, the headings have the following meanings: Dn = data register operand and M = memory operand.

Table 7-13. Multi-Precision Instruction Execution Times

Instruction	Size	op Dn, Dn	op M, M
ADDX	Byte, Word	4(1 0)	18(3 1)
	Long	8(1 0)	30(5 2)
CMPM	Byte, Word		12(3 0)
	Long		20(5 0)
SUBX	Byte, Word	4(1 0)	18(3 1)
	Long	8(1 0)	30(5 2)
ABCD	Byte	6(1 0)	18(3 1)
SBCD	Byte	6(1 0)	18(3 1)

7.2.11 Miscellaneous Instruction Execution Times

Tables 7-14 and 7-15 indicate the number of clock periods for the following miscellaneous instructions. The number of bus read and write cycles is shown in parenthesis as (r w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Table 7-14. Miscellaneous Instruction Execution Times

Instruction	Size	Register	Memory
ANDI to CCR	Byte	20(3 0)	
ANDI to SR	Word	20(3 0)	
CHK	–	10(1 0) +	
EORI to CCR	Byte	20(3 0)	
EORI to SR	Word	20(3 0)	
ORI to CCR	Byte	20(3 0)	
ORI to SR	Word	20(3 0)	
MOVE from SR	–	6(1 0)	8(1 1) +
MOVE to CCR	–	12(2 0)	12(2 0) +
MOVE to SR	–	12(2 0)	12(2 0) +
EXG	–	6(1 0)	
EXT	Word	4(1 0)	
	Long	4(1 0)	
LINK	–	16(2 2)	
MOVE from USP	–	4(1 0)	
MOVE to USP	–	4(1 0)	
NOP	–	4(1 0)	
RESET	–	132(1 0)	
RTE	–	20(5 0)	
RTR	–	20(5 0)	
RTS	–	16(4 0)	
STOP	–	4(0 0)	
SWAP	–	4(1 0)	
TRAPV	–	4(1 0)	
UNLK	–	12(3 0)	

+ add effective address calculation time

Table 7-15. Move Peripheral Instruction Execution Times

Instruction	Size	Register → Memory	Memory → Register
MOVEP	Word	16(2/2)	16(4/0)
	Long	24(2/4)	24(6/0)

7.2.12 Exception Processing Execution Times

Table 7-16 indicates the number of clock periods for exception processing. The number of clock periods includes the time for all stacking, the vector fetch, and the fetch of the first two instruction words of the handler routine. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 7-16. Exception Processing Execution Times

Exception	Periods
Address Error	**50**(4/7)
Bus Error	**50**(4/7)
CHK Instruction	**44**(5/4) +
Divide by Zero	**42**(5/4)
Illegal Instruction	**34**(4/3)
Interrupt	**44**(5/3) *
Privilege Violation	**34**(4/3)
$\overline{\text{RESET}}$ * *	**40**(6/0)
Trace	**34**(4/3)
TRAP Instruction	**38**(4/4)
TRAPV Instruction	**34**(4/3)

+ add effective address calculation time

* The interrupt acknowledge cycle is assumed to take four clock periods.

* * Indicates the time from when $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are first sampled as negated to when instruction execution starts.

SECTION 8
ELECTRICAL SPECIFICATIONS

This section contains electrical specifications and associated timing information for the TS68000

8.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range TS68000C TS68000V TS68000M	T_A	T_L to T_H 0 to 70 −40 to 85 −55 to 125	°C
Storage Temperature	T_{stg}	−55 to 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{CC}).

8.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Symbol	Value	Rating
Thermal Resistance (Still Air)					
Ceramic DIL	θ_{JA}	30	θ_{JC}	15*	°C/W
Ceramic PGA		33		15	
LCCC		50		25	
Plastic DIL		35		15*	
PLCC		60		25*	

*Estimated

8.3 DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 Vdc ±5%; GND = 0 Vdc; $T_A = T_L$ to T_H; see Figures 8-1, 8-2, and 8-3)

Characteristic		Symbol	Min	Max	Unit
Input High Voltage		V_{IH}	2.0	V_{CC}	V
Input Low Voltage		V_{IL}	GND−0.3	0.8	V
Input Leakage Current @ 5.25 V	$\overline{BERR}$, $\overline{BGACK}$, $\overline{BR}$, $\overline{DTACK}$, CLK, $\overline{IPL0}$-$\overline{IPL2}$, $\overline{VPA}$	I_{in}	−	2.5	μA
	$\overline{HALT}$, $\overline{RESET}$		−	20	
Three-State (Off State) Input Current @ 2.4 V/0.4 V	$\overline{AS}$, A1-A23, D0-D15, FC0-FC2, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$	I_{TSI}	−	20	μA
Output High Voltage (I_{OH} = −400 μA)	E*	V_{OH}	V_{CC}−0.75	−	V
	E, $\overline{AS}$, A1-A23, $\overline{BG}$, D0-D15, FC0-FC2, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$		2.4	−	
Output Low Voltage		V_{OL}			V
(I_{OL} = 1.6 mA)	$\overline{HALT}$		−	0.5	
(I_{OL} = 3.2 mA)	A1-A23, $\overline{BG}$, FC0-FC2		−	0.5	
(I_{OL} = 5.0 mA)	$\overline{RESET}$		−	0.5	
(I_{OL} = 5.3 mA)	E, $\overline{AS}$, D0-D15, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$		−	0.5	
Power Dissipation (See **8.4 POWER CONSIDERATIONS**)		P_D***	−	−	W
Capacitance (V_{in} = 0 V, T_A = 25°C; Frequency = 1 MHz)**		C_{in}	−	20.0	pF

*With external pullup resistor of 1.1 kΩ.

**Capacitance is periodically sampled rather than 100% tested.

***During normal operation instantaneous V_{CC} current requirements may be as high as 1.5 A

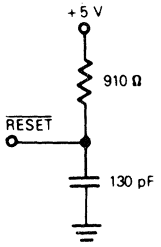


Figure 8-1. $\overline{\text{RESET}}$ Test Load

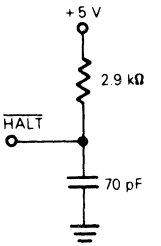


Figure 8-2. $\overline{\text{HALT}}$ Test Load

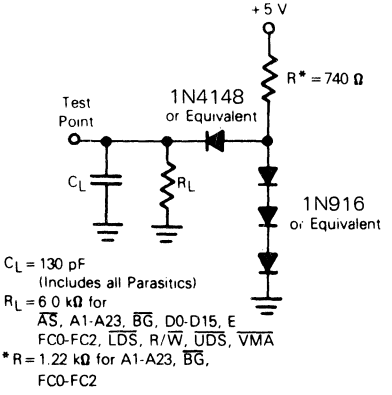


Figure 8-3. Test Loads

8.4 POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts – Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins – User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

Figure 8-4 illustrates the graphic solution to the equations, given above, for the specification power dissipations of 1.50 and 1.75 watts over the ambient temperature range of –55°C to 125°C using an average θ_{JA} of 40°C/watt to represent the various TS68000 packages. However, actual θ_{JA}'s in the range of 30°C to 50°C/watt only change the curves slightly.

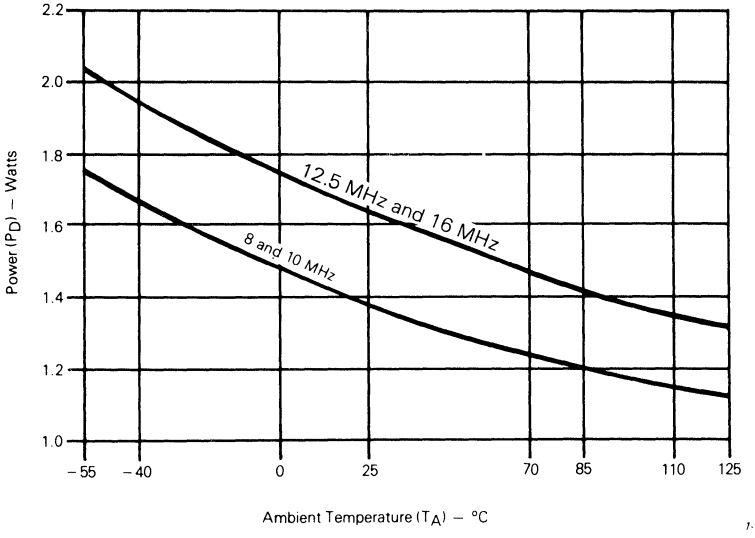


Figure 8-4. TS68000 Power Dissipation (P_D) vs Ambient Temperature (T_A)

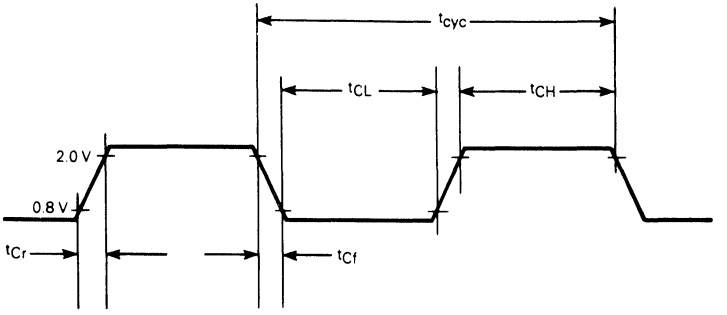
The total thermal resistance of a package (θ_{JA}) can be separated into two components, θ_{JC} and θ_{CA}, representing the barrier to heat flow from the semiconductor junction to the package (case) surface (θ_{JC}) and from the case to the outside ambient (θ_{CA}). These terms are related by the equation:

$$\theta_{JA} = \theta_{JC} + \theta_{CA} \quad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling and thermal convention. Thus, good thermal management on the part of the user can significantly reduce θ_{CA} so that θ_{JA} approximately equals θ_{JC}. Substitution of θ_{JC} for θ_{JA} in equation (1) will result in a lower semiconductor junction temperature.

8.5 AC ELECTRICAL SPECIFICATIONS — CLOCK TIMING (See Figure 8-5)

Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Frequency of Operation	f	4.0	8.0	4.0	10.0	4.0	12.5	4.0	16	MHz
Cycle Time	t_{cyc}	125	250	100	250	80	250	60	250	ns
Clock Pulse Width	t_{CL} t_{CH}	55 55	125 125	45 45	125 125	35 35	125 125	25 25	125 125	ns
Rise and Fall Times	t_{Cr} t_{Cf}	– –	10 10	– –	10 10	– –	5 5	– –	5 5	ns



NOTE:
Timing measurements are referenced to and from a low voltage of 0.8 volt and high a voltage of 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8 volt and 2.0 volts.

Figure 8-5. Clock Input Timing Diagram

8.6 AC ELECTRICAL SPECIFICATIONS — READ CYCLES

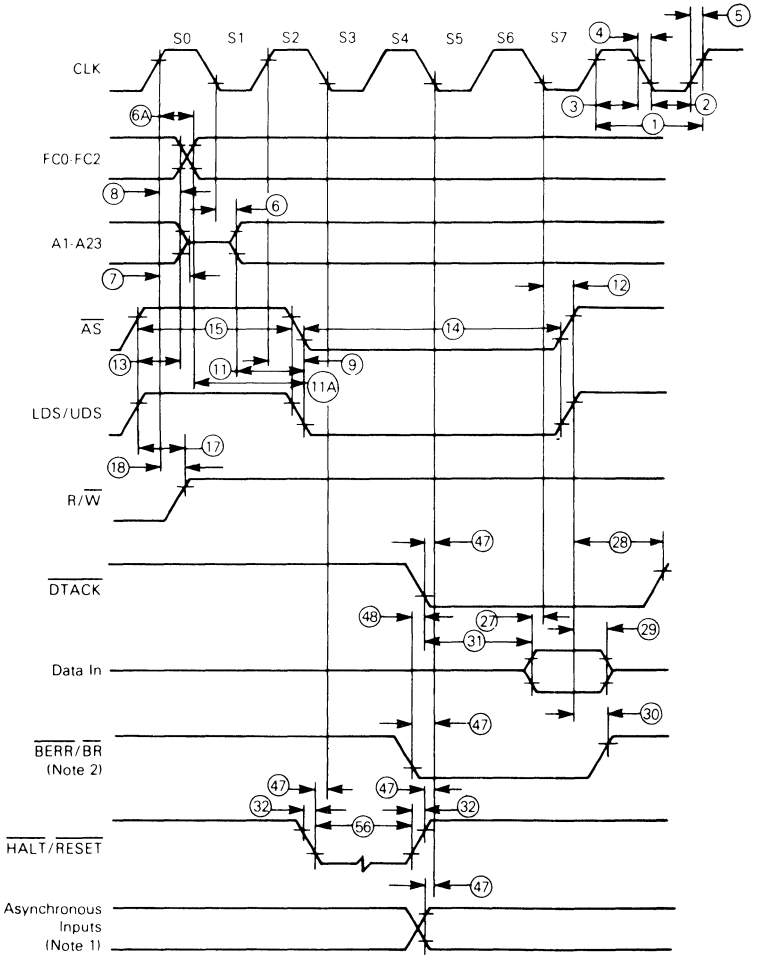
(V_{CC}=5.0 Vdc ± 5 % ; GND=0 Vdc ; T_A=T_L to T_H ; see Figure 8-6)

Num.	Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		16 MHz		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
1	Clock Period	tCYC	125	250	100	250	80	250	60	250	ns
2	Clock Width Low	tCL	55	125	45	125	35	125	25	125	ns
3	Clock Width High	tCH	55	125	45	125	35	125	25	125	ns
4	Clock Fall Time	tCf	–	10	–	10	–	5	–	5	ns
5	Clock Rise Time	tCr	–	10	–	10	–	5	–	5	ns
6	Clock Low to Address Valid	tCLAV	–	70	–	60	–	55	–	50	ns
6A	Clock High to FC Valid	tCHFCV	–	70	–	60	–	55	–	50	ns
7	Clock High to Address, Data Bus High Impedance (Maximum)	tCHADZ	–	80	–	70	–	60	–	55	ns
8	Clock High to Address, FC Invalid (Minimum)	tCHAFI	0	–	0	–	0	–	0	–	ns
9[1]	Clock High to $\overline{AS}$, $\overline{DS}$ Low	tCHSL	0	60	0	55	0	55	0	55	ns
11[2]	Address Valid to $\overline{AS}$, $\overline{DS}$ Low	tAVSL	30	–	20	–	0	–	0	–	ns
11A[2]	FC Valid to $\overline{AS}$, $\overline{DS}$ Low	tFCVSL	60	–	50	–	40	–	30	–	ns
12[1]	Clock Low to $\overline{AS}$, $\overline{DS}$ High	tCLSH	–	70	–	55	–	50	–	45	ns
13[2]	$\overline{AS}$, $\overline{DS}$ High to Address/FC Invalid	tSHAFI	30	–	20	–	10	–	10	–	ns
14[2]	$\overline{AS}$, $\overline{DS}$ Width Low	tSL	240	–	195	–	160	–	120	–	ns
15[2]	$\overline{AS}$, $\overline{DS}$ Width High	tSH	150	–	105	–	65	–	50	–	ns
17[2]	$\overline{AS}$, $\overline{DS}$ High to R/$\overline{W}$ High	tSHRH	40	–	20	–	10	–	10	–	ns
18[1]	Clock High to R/$\overline{W}$ High	tCHRH	0	70	0	60	0	60	0	55	ns
27[5]	Data In to Clock Low (Setup Time)	tDICL	15	–	10	–	10	–	10	–	ns
28[2]	$\overline{AS}$, $\overline{DS}$ High to $\overline{DTACK}$ High	tSHDAH	0	245	0	190	0	150	0	110	ns
29	$\overline{AS}$, $\overline{DS}$ High to Data In Invalid (Hold Time)	tSHDII	0	–	0	–	0	–	0	–	ns
30	$\overline{AS}$, $\overline{DS}$ High to $\overline{BERR}$ High	tSHBEH	0	–	0	–	0	–	0	–	ns
31[2,5]	$\overline{DTACK}$ Low to Data In (Setup Time)	tDALDI	–	90	–	65	–	50	–	30	ns
32	$\overline{HALT}$ and $\overline{RESET}$ Input Transition Time	tRHr,f	0	200	0	200	0	200	0	200	ns
47[5]	Asynchronous Input Setup Time	tASI	20	–	20	–	20	–	20	–	ns
48[3]	$\overline{BERR}$ Low to $\overline{DTACK}$ Low	tBELDAL	20	–	20	–	20	–	20	–	ns
56[4]	$\overline{HALT}$ / $\overline{RESET}$ Pulse Width	tHRPW	10	–	10	–	10	–	10	–	Clk.Per.

NOTES :

1. For a loading capacitance of less than or equal to 50 picofarads, subtract 5 nanoseconds from the value given in the maximum columns.
2. Actual value depends on clock period.
3. If 47 is satisfied for both $\overline{DTACK}$ and $\overline{BERR}$, 48 may be 0 nanosecond.
4. For power up, the MPU must be held in $\overline{RESET}$ state for 100 milliseconds to allow stabilization of on-chip circuitry. After the system is powered up, 56 refers to the minimum pulse width required to reset the system.
5. If the asynchronous setup time (47) requirements are satisfied, the $\overline{DTACK}$ low-to-data setup time (31) requirement can be ignored. The data must only satisfy the data-in to clock-low setup time (27) for the following cycle.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.



NOTES:

1. Setup time for the synchronous inputs BGACK, IPL0-2, and VPA guarantees their recognition at the next falling edge of the clock
2. BR need fall at this time only in order to insure being recognized at the end of this bus cycle.
3. Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8 volt and 2.0 volts.

Figure 8-6. Read Cycle Timing Diagram

8.6 AC ELECTRICAL SPECIFICATIONS — WRITE CYCLES

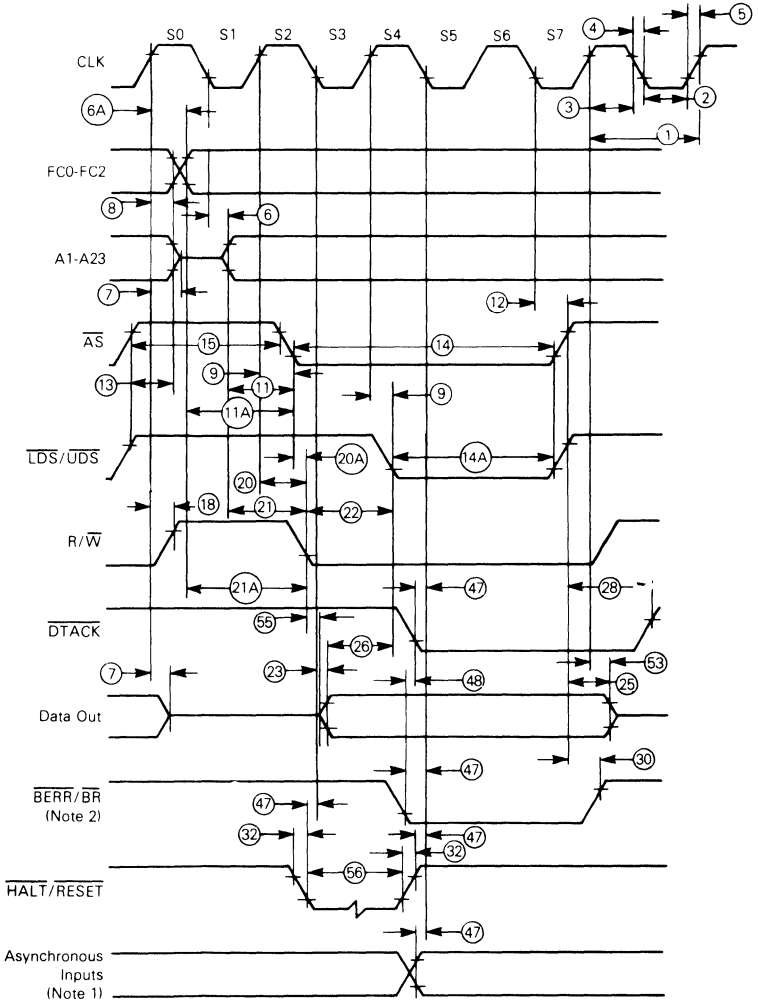
(V_{CC}=5.0 Vdc ± 5 % ; GND=0 Vdc ; T_A=T_L to T_H ; see Figure 8-7)

Num.	Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		16 MHz		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
1	Clock Period	t_{cyc}	125	250	100	250	80	250	60	250	ns
2	Clock Width Low	t_{CL}	55	125	45	125	35	125	25	125	ns
3	Clock Width High	t_{CH}	55	125	45	125	35	125	25	125	ns
4	Clock Fall Time	t_{Cf}	–	10	–	10	–	5	–	5	ns
5	Clock Rise Time	t_{Cr}	–	10	–	10	–	5	–	5	ns
6	Clock Low to Address Valid	t_{CLAV}	–	70	–	60	–	55	–	50	ns
6A	Clock High to FC Valid	t_{CHFCV}	–	70	–	60	–	55	–	50	ns
7	Clock High to Address, Data Bus High Impedance (Maximum)	t_{CHADZ}	–	80	–	70	–	60	–	55	ns
8	Clock High to Address, FC Invalid (Minimum)	t_{CHAFI}	0	–	0	–	0	–	0	–	ns
9[1]	Clock High to $\overline{AS}$, $\overline{DS}$ Low	t_{CHSL}	0	60	0	55	0	55	0	55	ns
11[2]	Address Valid to $\overline{AS}$ Low	t_{AVSL}	30	–	20	–	0	–	5	–	ns
11A[2]	FC Valid to $\overline{AS}$ Low	t_{FCVSL}	60	–	50	–	40	–	30	–	ns
12[1]	Clock Low to $\overline{AS}$, $\overline{DS}$ High	t_{CLSH}	–	70	–	55	–	50	–	45	ns
13[2]	$\overline{AS}$, $\overline{DS}$ High to Address/FC Invalid	t_{SHAFI}	30	–	20	–	10	–	10	–	ns
14[2]	$\overline{AS}$ Low	t_{SL}	240	–	195	–	160	–	120	–	ns
14A[2]	$\overline{DS}$ Width Low	t_{DSL}	115	–	95	–	80	–	60	–	ns
15[2]	$\overline{AS}$, $\overline{DS}$ Width High	t_{SH}	150	–	105	–	65	–	50	–	ns
18[1]	Clock High to R/$\overline{W}$ High	t_{CHRH}	0	70	0	60	0	60	0	55	ns
20[1]	Clock High to R/$\overline{W}$ Low	t_{CHRL}	–	70	–	60	–	60	–	55	ns
20A[6]	$\overline{AS}$, Low to R/$\overline{W}$ Valid	t_{ASRV}	–	20	–	20	–	20	–	20	ns
21[2]	Address Valid to R/$\overline{W}$ Low	t_{AVRL}	20	–	0	–	0	–	0	–	ns
21A[2]	FC Valid to R/$\overline{W}$ Low	t_{FCVRL}	60	–	50	–	30	–	20	–	ns
22[2]	R/$\overline{W}$ Low to $\overline{DS}$ Low	t_{RLSL}	80	–	50	–	30	–	20	–	ns
23	Clock Low to Data Out Valid	t_{CLDO}	–	70	–	55	–	55	–	50	ns
25[2]	$\overline{AS}$, $\overline{DS}$ High to Data Out Invalid	t_{SHDOI}	30	–	20	–	15	–	10	–	ns
26[2]	Data Out Valid to $\overline{DS}$ Low	t_{DOSL}	30	–	20	–	15	–	15	–	ns
28[2]	$\overline{AS}$, $\overline{DS}$ High to $\overline{DTACK}$ High	t_{SHDAH}	0	245	0	190	0	150	0	110	ns
30	$\overline{AS}$, $\overline{DS}$ High to $\overline{BERR}$ High	t_{SHBEH}	0	–	0	–	0	–	0	–	ns
32	$\overline{HALT}$ and $\overline{RESET}$ Input Transition Time	$t_{RHr,f}$	0	200	0	200	0	200	0	200	ns
47[5]	Asynchronous Input Setup Time	t_{ASI}	20	–	20	–	20	–	20	–	ns
48[3]	$\overline{BERR}$ Low to $\overline{DTACK}$ Low	t_{BELDAL}	20	–	20	–	20	–	20	–	ns
53	Clock High to Data Out Invalid	t_{CHDOI}	0	–	0	–	0	–	0	–	ns
55	R/$\overline{W}$ to Data Bus Driven	t_{RLDBD}	30	–	20	–	10	–	10	–	ns
56[4]	$\overline{HALT}$ / $\overline{RESET}$ Pulse Width	t_{HRPW}	10	–	10	–	10	–	10	–	Clk.Per.

NOTES :

1. For a loading capacitance of less than or equal to 50 picofarads, subtract 5 nanoseconds from the value given in the maximum columns
2. Actual value depends on clock period.
3. If 47 is satisfied for both $\overline{DTACK}$ and $\overline{BERR}$, 48 may be 0 nanoseconds.
4. For power up, the MPU must be held in $\overline{RESET}$ state for 100 milliseconds to allow stabilization of on-chip circuitry. After the syster is powered up, 56 refers to the minimum pulse width required to reset the system.
5. If the asynchronous setup time (47) requirements are satisfied, the $\overline{DTACK}$ low-to-data setup time (31) requirement can be ignorec The data must only satisfy the data-in to clock-low setup time (27) for the following cycle.
6. When $\overline{AS}$, and R/$\overline{W}$ are equally loaded (±20 %), Subtract 10 nanoseconds from the values in these columns.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.



NOTES:

1. Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8 volt and 2.0 volts.
2. Because of loading variations, R/$\overline{W}$ may be valid after $\overline{AS}$ even though both are initiated by the rising edge of S2 (Specification 20A).

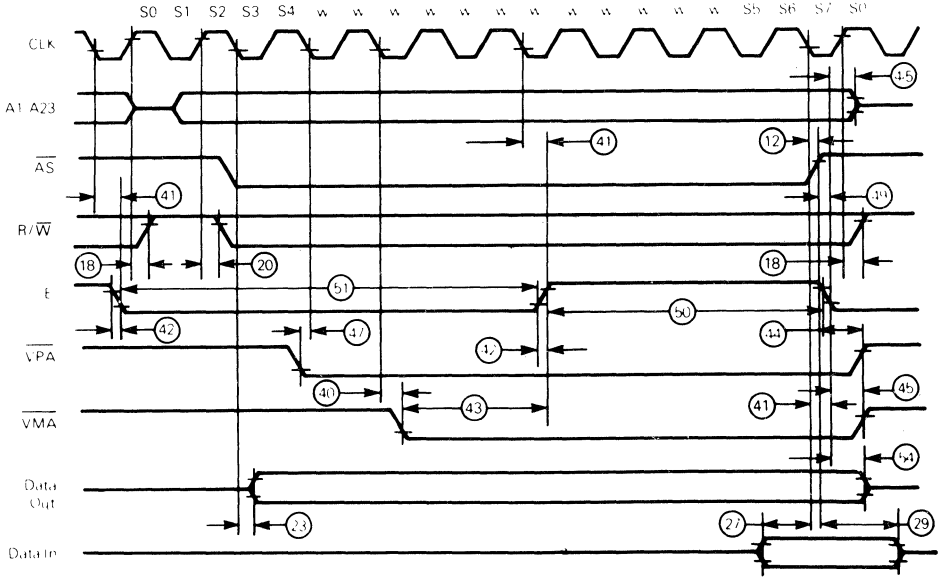
Figure 8-7. Write Cycle Timing Diagram

8.7 AC ELECTRICAL SPECIFICATIONS — TS 68000 to 6800 PERIPHERAL

(V_{CC}=5.0 Vdc ± 5 %; GND=0 Vdc; T_A=T_L to T_H; refer to Figures 8-8 and 8-9)

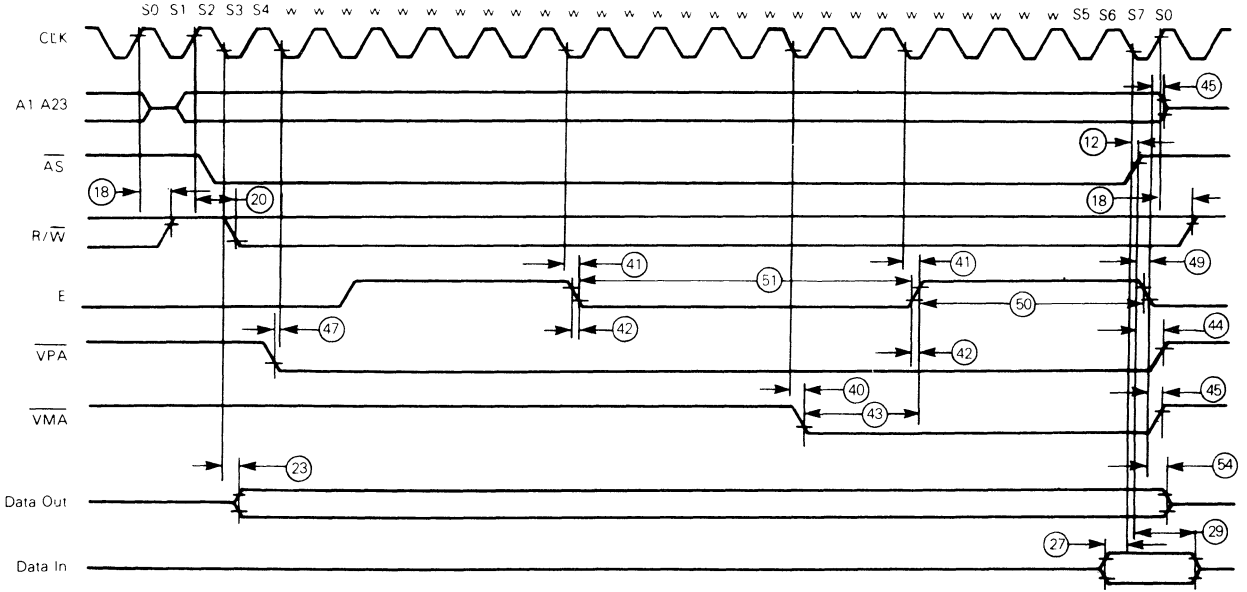
Num.	Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		16 MHz		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
12[1]	Clock Low to AS, DS High	tCLSH	–	70	–	55	–	50	–	45	ns
18[1]	Clock High to R/W High	tCHRH	0	70	0	60	0	60	0	55	ns
20[1]	Clock High to R/W Low (Write)	tCHRL	–	70	–	60	–	60	–	55	ns
23	Clock Low to Data Out Valid (Write)	tCLDO	–	70	–	55	–	55	–	50	ns
27[2]	Data In to Clock Low (Setup Time on Read)	tCLDO	15	–	10	–	10	–	10	–	ns
29	AS, DS High to Data In Invalid (Hold Time on Read)	tSHDII	0	–	0	–	0	–	0	–	ns
40	Clock Low to VMA Low	tCLVML	–	70	–	70	–	70	–	70	ns
41	Clock Low to E Transition	tCLET	–	70	–	55	–	45	–	45	ns
42	E Output Rise and Fall Time	tEr,f	–	25	–	25	–	25	–	25	ns
43	VMA Low to E High	tVMLEH	200	–	150	–	90	–	70	–	ns
44	AS, DS High to VPA High	tSHVPH	0	120	0	90	0	70	0	50	ns
45	E Low to Control, Address Bus Invalid (Address Hold Time)	tELCAI	30	–	10	–	10	–	10	–	ns
47[2]	Asynchronous Input Setup Time	tASI	20	–	20	–	20	–	20	–	ns
49[3]	AS, DS High to E Low	tSHEL	−70	70	−55	55	−45	45	−45	+45	ns
50	E Width High	tEH	450	–	350	–	280	–	210	–	ns
51	E Width Low	tEL	700	–	550	–	440	–	330	–	ns
54	E Low to Data Out Invalid	tELDOI	30	–	20	–	15	–	10	–	ns

NOTES:
1. For a loading capacitance of less than or equal to 50 picofarads, subtract 5 nanoseconds from the value given in the maximum columns.
2. If the asynchronous setup time (47) requirements are satisfied, the DTACK low-to-data setup time (31) requirement can be ignored. The data must only satisfy the date-in clock-low setup time (27) for the following cycle.
3. The falling edge of S6 triggers both the negation of the strobes (AS, and x DS) and the falling edge of E. Either of these events can occur first, depending upon the loading on each signal. Specification 49 indicates the absolute maximum skew that will occur between the rising edge of the strobes and the falling edge of the E clock.



This timing diagram is included for those who wish to design their own circuit to generate VMA it shows the best case possibly attainable

Figure 8-8. TS 68000 to 6800 Peripheral Timing Diagram - Best Case



NOTE: This timing diagram is included for those who wish to design their own circuit to generate VMA. It shows the worst case possibly attainable.

Figure 8-9. TS68000 to 6800 Peripheral Timing Diagram — Worst Case

8.8 AC ELECTRICAL SPECIFICATIONS — BUS ARBITRATION

(V_{CC}=5.0 Vdc ± 5 %; GND=0 Vdc; T_A=T_L to T_H; see Figures 8-10, 8-11, and 8-12)

Num.	Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		16 MHz		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
7	Clock High to Address, Data Bus High Impedance	tCHADZ	–	80	–	70	–	60	–	55	ns
16	Clock High to Control Bus High Impedance	tCHCZ	–	80	–	70	–	60	–	55	ns
33	Clock High to $\overline{BG}$ Low	tCHGL	–	70	–	60	–	50	–	45	ns
34	Clock High to $\overline{BG}$ High	tCHGH	–	70	–	60	–	50	–	45	ns
35	$\overline{BR}$, Low to $\overline{BG}$ Low	tBRLGL	1.5	90 ns + 3.5	1.5	80 ns + 3.5	1.5	70 ns + 3.5	1.5	65 ns + 3.5	Clk.Per.
36[1]	$\overline{BR}$ High to $\overline{BG}$ High	tBKHGH	1.5	90 ns + 3.5	1.5	80 ns + 3.5	1.5	70 ns + 3.5	1.5	65 ns + 3.5	Clk.Per.
37	$\overline{BGACK}$ Low to $\overline{BG}$ High	tGALGH	1.5	90 ns + 3.5	1.5	80 ns + 3.5	1.5	70 ns + 3.5	1.5	65 ns + 3.5	Clk.Per.
37A[2]	$\overline{BGACK}$ Low to $\overline{BR}$ High	tGALBRH	20	1.5 Clocks	20	1.5 Clocks	20	1.5 Clocks	20	1.5 Clocks	ns
38	$\overline{BG}$ Low to Control, Address, Data Bus High Impedance ($\overline{AS}$ High)	tGLZ	–	80	–	70	–	60	–	55	ns
39	$\overline{BG}$ Width High	tGH	1.5	–	1.5	–	1.5	–	1.5	–	Clk.Per.
46	$\overline{BGACK}$ Width Low	tGAL	1.5	–	1.5	–	1.5	–	1.5	–	Clk.Per.
47[3]	Asynchronous Input Setup Time	tASI	20	–	20	–	20	–	20	–	ns
57	$\overline{BGACK}$ High to Control Bus Driven $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$	tGABD	1.5	–	1.5	–	1.5	–	–	1.5	Clk.Per.
	FC_x, R/$\overline{W}$, $\overline{VMA}$		1		1		1			1	
58[1]	$\overline{BR}$ High to Control Bus Driven	tRHBD	1.5	–	1.5	–	1.5	–	–	1.5	Clk.Per.

NOTES:
1. The processor will negate $\overline{BG}$ and begin driving the bus again if external arbitration logic negates $\overline{BR}$ before asserting $\overline{BGACK}$
2. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{BG}$ may be reasserted.
3. If the asynchronous setup time (47) requirements are satisfied, the $\overline{DTACK}$ low-to-data setup time (31) requirement can ban ignored. The data must only satisfy the date-in clock-low setup time (27) for the following cycle.

Figures 8-10, 8-11, and 8-12 depict the three bus arbitration cases that can arise. Figure 8-10 shows the timing where $\overline{AS}$ is negated when the processor asserts $\overline{BG}$ (Idle Bus Case). Figure 8-11 shows the timing where $\overline{AS}$ is asserted when the processor asserts $\overline{BG}$ (Active Bus Case). Figure 8-12 shows the timing where more than one bus master are requesting the bus. Refer to **4.2.2 Bus Arbitration** for a complete discussion of bus arbitration.

The waveforms shown in Figures 8-10, 8-11, and 8-12 should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

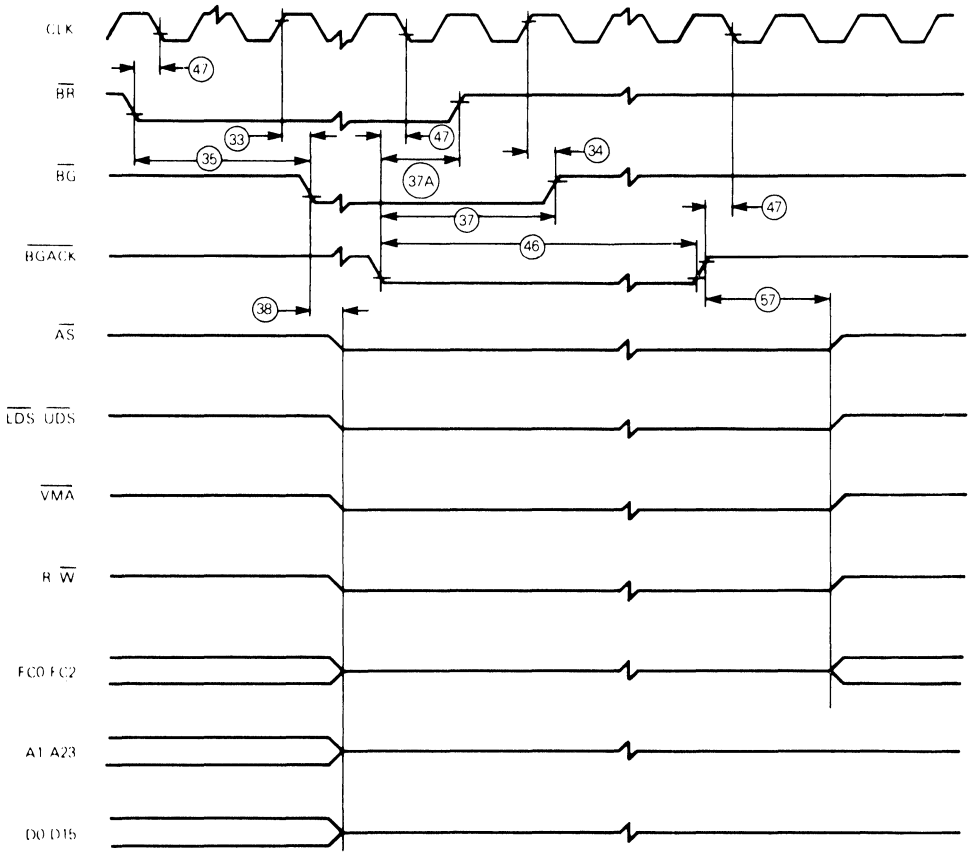


Figure 8-10. Bus Arbitration Timing Diagram — Idle Bus Case

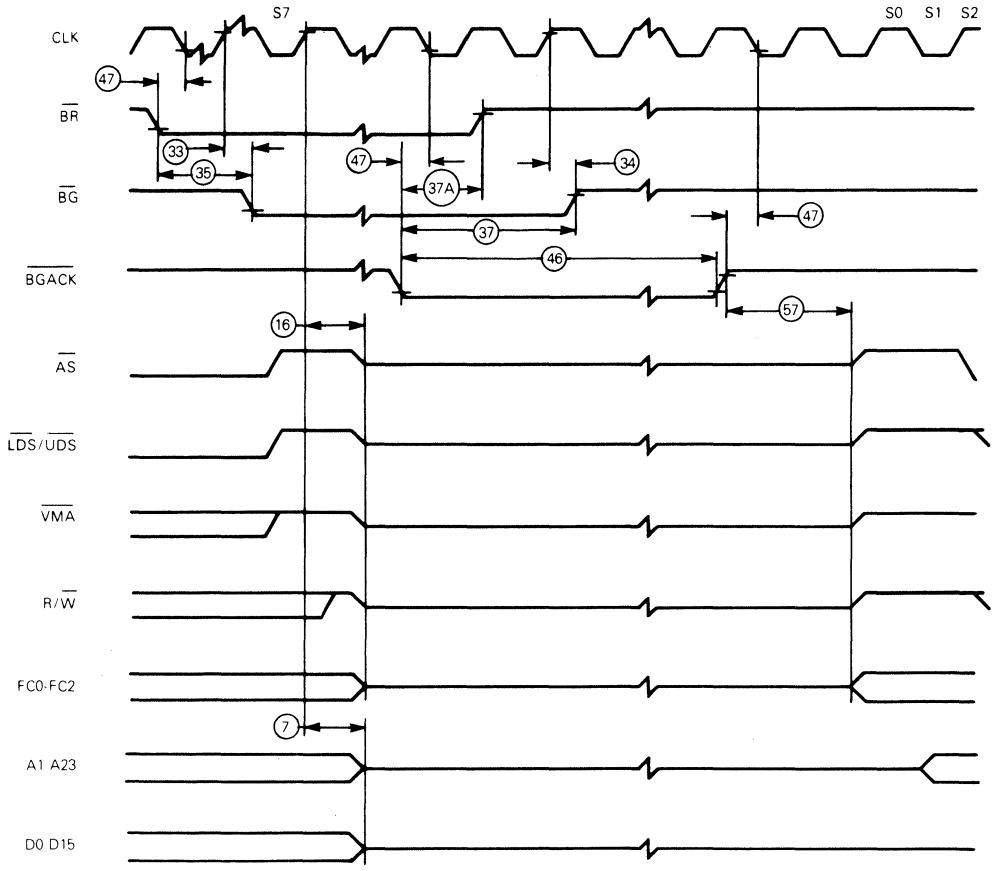


Figure 8-11. Bus Arbitration Timing Diagram — Active Bus Case

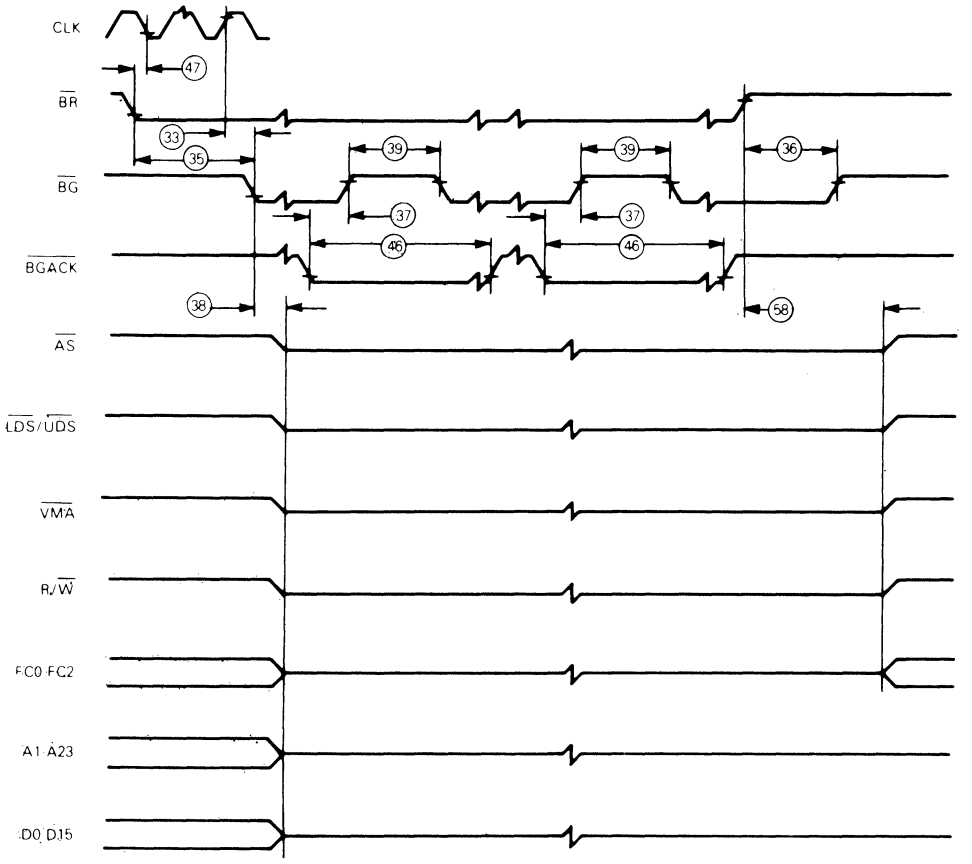


Figure 8-12. Bus Arbitration Timing Diagram — Multiple Bus Requests

SECTION 9
ORDERING INFORMATION

This section contains detailed information to be used as a guide when ordering the TS 68000

9.1. STANDARD VERSIONS

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic DIL	8.0	0°C to + 70°C	TS 68000 CC8
C Suffix	8.0	−40°C to + 85°C	TS 68000 VC8
	8.0	−55°C to + 125°C	TS 68000 MC8
	10.0	0°C to + 70°C	TS 68000 CC10
	10.0	−40°C to + 85°C	TS 68000 VC10
	10.0	−55°C to + 125°C	TS 68000 MC10
	12.5	0°C to + 70°C	TS 68000 CC12
	12.5	−40°C to + 85°C	TS 68000 VC12
	12.5	−55°C to + 125°C	TS 68000 MC12
	16.0	0°C to + 70°C	TS 68000 CC16
Plastic DIL	8.0	0°C to + 70°C	TS 68000 CP8
P Suffix	10.0	0°C to + 70°C	TS 68000 CP10
	12.5	0°C to + 70°C	TS 68000 CP12
	16.0	0°C to + 70°C	TS 68000 CP16
PLCC	8.0	0°C to + 70°C	TS 68000 CFN8
FN Suffix	10.0	0°C to + 70°C	TS 68000 CFN10
	12.5	0°C to + 70°C	TS 68000 CFN12
	16.0	0°C to + 70°C	TS 68000 CFN16
LCCC	8.0	−55°C to + 125°C	TS 68000 ME8
E Suffix	10.0	−55°C to + 125°C	TS 68000 ME10
	12.5	−55°C to + 125°C	TS 68000 ME 12
Pin Grid Array	8.0	0°C to + 70°C	TS 68000 CR8
R Suffix	8.0	−55°C to + 125°C	TS 68000 MR8
	10.0	0°C to + 70°C	TS 68000 CR10
	10.0	−55°C to + 125°C	TS 68000 MR10
	12.5	0°C to + 70°C	TS 68000 CR12
	12.5	−55°C to + 125°C	TS 68000 MR12
	16.0	0°C to + 70°C	TS 68000 CR16

9.2. HI-REL VERSIONS

In order to fit more closely to customer specific requirements, THOMSON SEMICONDUCTEURS is proposing different screening levels for its HI-REL ranges.

G/B screening : Available only from THOMSON SEMICONDUCTEURS, this quality level, very close to the MIL-STD-883, is a cost effective alternative for customers who want to buy HI-REL devices (low guaranteed AQL). The G/B level is in full accordance with the NFC 96883 class G.

B/B screening : Full accordance with the MIL-STD-883 Rev.C, class B (US), the CECC 90.000, class B (European) and with the NFC 96883 class B (French).

Details on screening procedures for these levels of selection are available on request (please contact our sales representatives).

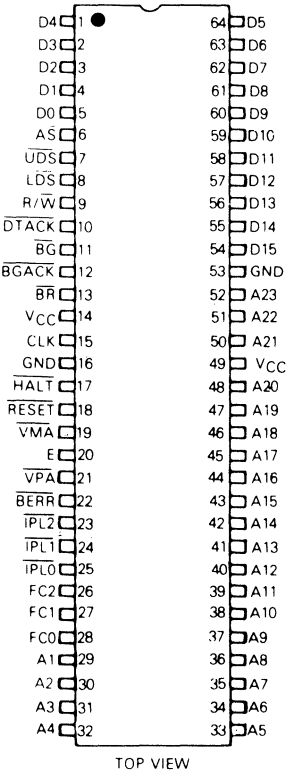
Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic DIL	8.0	−55°C to + 125°C	TS 68000 MC G/B8
C Suffix	8.0	−55°C to + 125°C	TS 68000 MC B/B8
	10.0	−55°C to + 125°C	TS 68000 MC G/B10
	10.0	−55°C to + 125°C	TS 68000 MC B/B10
	12.5	−55°C to + 125°C	TS 68000 MC G/B12
	12.5	−55°C to + 125°C	TS 68000 MC B/B12
LCCC	8.0	−55°C to + 125°C	TS 68000 MEG/B8
E Suffix	8.0	−55°C to + 125°C	TS 68000 ME B/B8
	10.0	−55°C to + 125°C	TS 68000 ME G/B10
	10.0	−55°C to + 1256C	TS 68000 ME B/B10
	12.5	−55°C to + 125°C	TS 68000 ME G/B12
	12.5	−55°C to + 125°C	TS 68000 ME B/B12
Pin Grid Array	8.0	−55°C to + 125°C	TS 68000 MR G/B8
R Suffix	8.0	−55°C to + 125°C	TS 68000 MR B/B8
	10.0	−55°C to + 125°C	TS 68000 MR G/B10
	10.0	−55°C to + 125°C	TS 68000 MR B/B10
	12.5	−55°C to + 125°C	TS 68000 MR G/B12
	12.5	−55°C to + 125°C	TS 68000 MR B/B12

SECTION 10
MECHANICAL DATA

This section contains the pin assignments and package dimensions for the 64-pin dual-in-line, the 68-terminal chip carrier (LCCC), the 68-pin grid array, and the 68-pin quad pack (PLCC), versions of the TS68000.

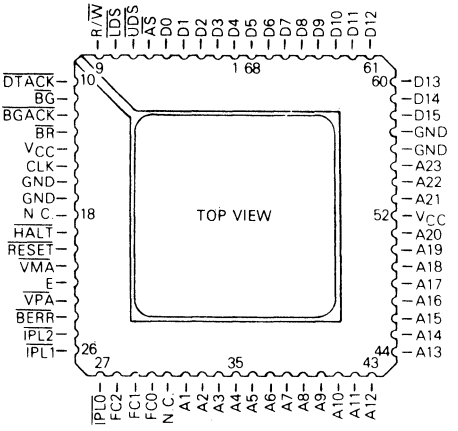
10.1 PIN ASSIGNMENTS

64-Pin Dual-in-Line Package

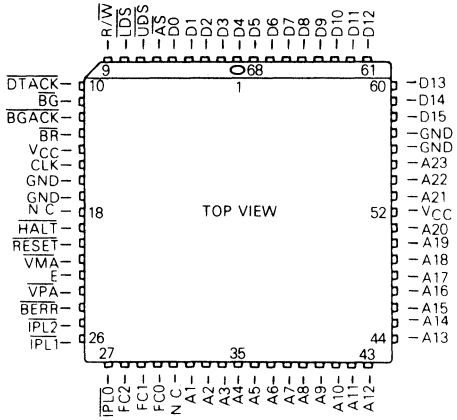


TOP VIEW

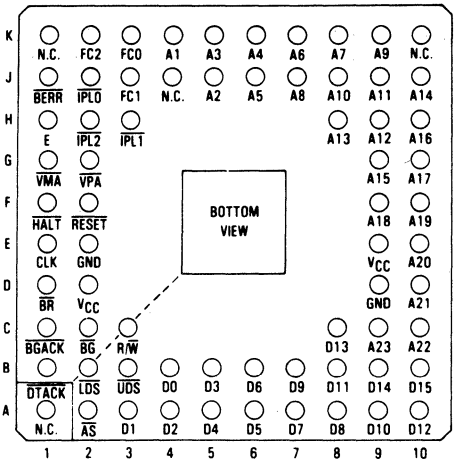
68-Terminal Chip Carrier (LCCC)



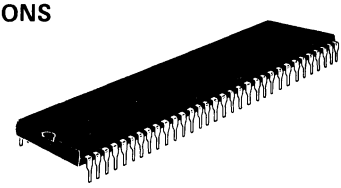
68-Pin Quad Pack (PLCC)



68-Pin Grid Array

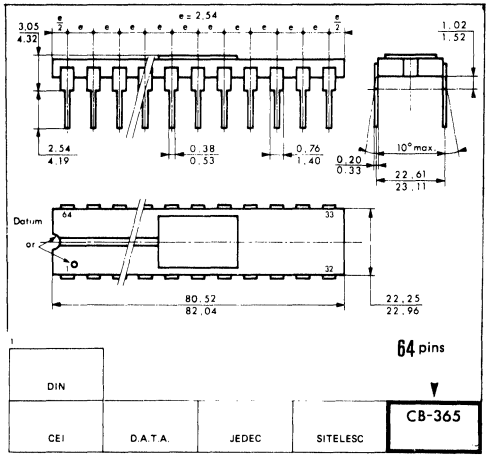


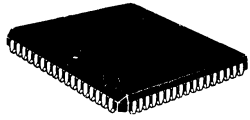
10.2 PACKAGE DIMENSIONS



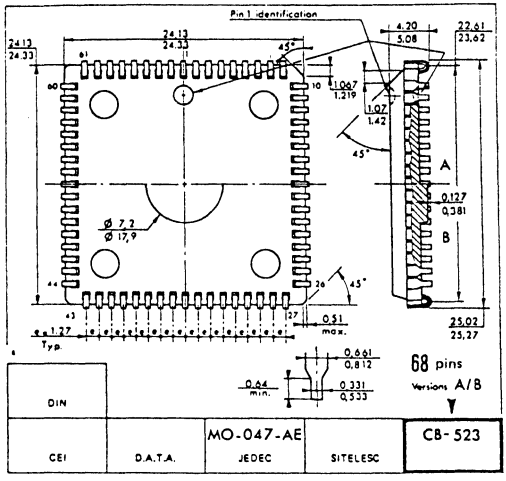
P SUFFIX
PLASTIC PACKAGE

C SUFFIX
CERAMIC PACKAGE

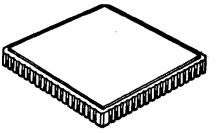




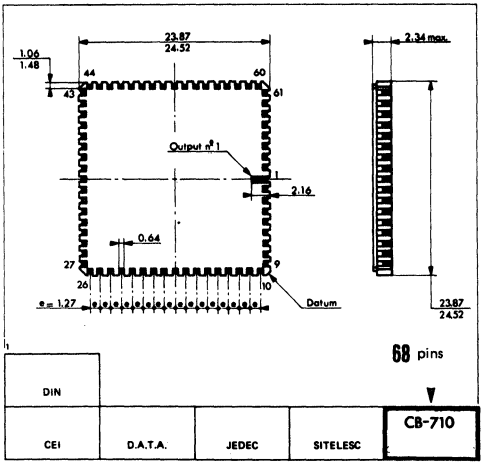
FN SUFFIX
PLCC 68



10.2 PACKAGE DIMENSIONS



E SUFFIX
LCCC 68



THOMSON COMPONENTS MOSTEK

TS68008

8/32-BIT MICROPROCESSOR WITH 8-BIT DATA BUS

ADVANCE INFORMATION

The TS68008 is a member of the TS68000 family of advanced microprocessors. This device allows the design of cost effective systems using 8-bit data buses while providing the benefits of a 32-bit microprocessor architecture. The performance of the TS68008 is greater than any 8-bit microprocessor and superior to several 16-bit microprocessors.

- 17 32-Bit Data and Address Registers
- 56 Basic Instruction Types
- Extensive Exception Processing
- Memory Mapped I/O
- 14 Addressing Modes
- 1 Mbyte Linear Addressing Space
- Complete Code Compatibility with the TS68000.

A system implementation based on an 8-bit data bus reduces system cost in comparison to 16-bit systems due to a more effective use of components and the fact that byte-wide memories and peripherals can be used much more effectively. In addition, the non-multiplexed address and data buses eliminate the need for external demultiplexers, thus further simplifying the system.

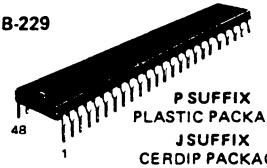
HMOS

HIGH-DENSITY, N-CHANNEL, SILICON-GATE, DEPLETION LOAD

8-/32-BIT MICROPROCESSOR WITH 8-BIT DATA BUS

CASES

CB-229

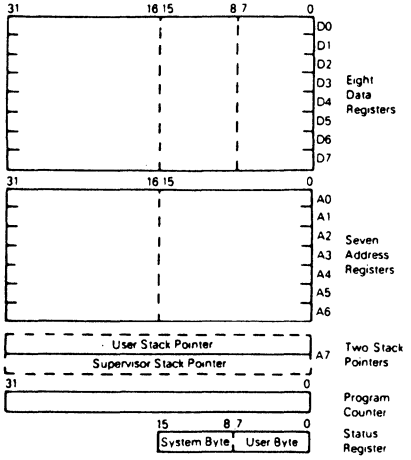


P SUFFIX
PLASTIC PACKAGE
J SUFFIX
CERDIP PACKAGE
C SUFFIX
CERAMIC PACKAGE

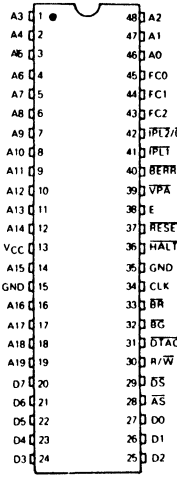
CB-522

FN SUFFIX
PLCC 52

PROGRAMMING MODEL



PIN ASSIGNMENT



SECTION 1
INTRODUCTION

The TS68008 is a member of the 68000 Family of advanced microprocessors. This device allows the design of cost effective systems using 8-bit data buses while providing the benefits of a 32-bit microprocessor architecture. The performance of the TS68008 is greater than any 8-bit microprocessor and superior to several 16-bit microprocessors.

The resources available to the TS68008 user consist of the following:

- 17 32-Bit Data and Address Registers
- 56 Basic Instruction Types
- Extensive Exception Processing
- Memory Mapped I/O
- 14 Addressing Modes
- Complete Code Compatibility with the TS68000

A system implementation based on an 8-bit data bus reduces system cost in comparison to 16-bit systems due to a more effective use of components and the fact that byte-wide memories and peripherals can be used much more effectively. In addition, the non-multiplexed address and data buses eliminate the need for external demultiplexers, thus further simplifying the system.

The TS68008 has full code compatibility (source and object) with the TS68000 which allows programs to be run on either MPU, depending on performance requirements and cost objectives.

The TS68008 is available in a 48-pin dual-in-line package (plastic or ceramic) and a 52-pin quad plastic package. Among the four additional pins of the 52-pin package, two additional address lines are included beyond the 20 address lines of the 48-pin package. The address range of the TS68008 is one or four megabytes with the 48- or 52-pin package, respectively.

The large non-segmented linear address space of the TS68008 allows large modular programs to be developed and executed efficiently. A large linear address space allows program segment sizes to be determined by the application rather than forcing the designer to adopt an arbitrary segment size without regard to the application's individual requirements.

The programmer's model is identical to that of the TS68000, as shown in Figure 1-1, with seventeen 32-bit registers, a 32-bit program counter, and a 16-bit status register. The first eight registers (D0-D7) are used as data registers for byte (8-bit), word (16-bit), and long word (32-bit) operations. The second set of seven registers (A0-A6), the user stack pointer (A7), and the system stack pointer (A7') may be used as software stack pointers and base address registers. In addition, the registers may be used for some simple word and long word data operations. All of the 17 registers may be used as index registers.

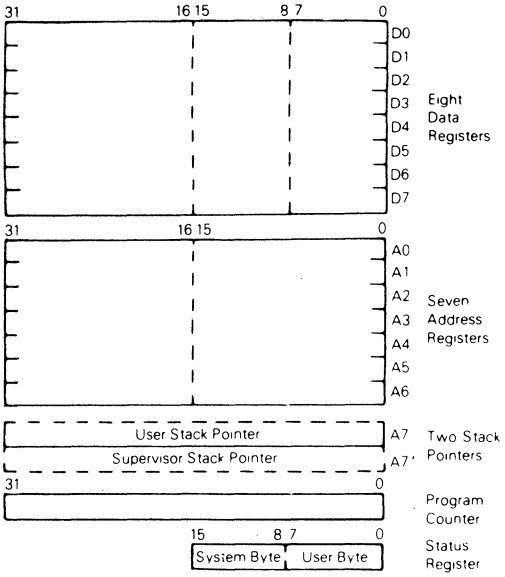


Figure 1-1. Programming Model

While all of the address registers can be used to create stacks and queues, the A7 address register, by convention, is used as the system stack pointer. Supplementing this convention is another address register, A7', also referred to as the system stack pointer. This powerful concept allows the supervisor mode and user mode of the TS68008 to each have their own system stack pointer (consistently referred to as SP) without needing to move pointers for each context of use when the mode is switched.

The system stack pointer (SP) is either the supervisor stack pointer (A7' ≡ SSP) or the user stack point (A7 ≡ USP), depending on the state of the S bit in the status register. If the S bit is set, indicating that the processor is in the supervisor state, when the SSP is the active system stack pointer and the USP is not used. If the S bit is clear, indicating that the processor is in the user state, then the USP is the active system stack pointer and the SSP is protected from user modification.

The status register, shown in Figure 1-2, may be considered as two bytes: the user byte and the system byte. The user byte contains five bits defining the overflow (V), zero (Z), negative (N), carry (C), and extended (X) condition codes. The system byte contains five defined bits. Three bits are used to define the current interrupt priority; any interrupt level higher than the current mask level will be recognized. (Note that level 7 interrupts are non-maskable – that is, level 7 interrupts are always processed.) Two additional bits indicate whether the processor is in the trace (T) mode and/or in the supervisor (S) state..

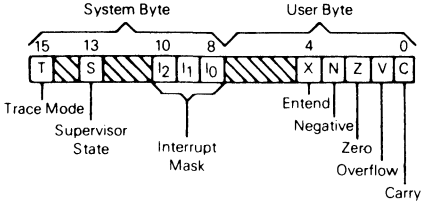


Figure 1-2. Status Register

1.1 DATA TYPES AND ADDRESSING MODES

Five basic data types are supported. These data types are:

- Bits
- BCD Digits (4 bits)
- Bytes (8 bits)
- Words (16 bits)
- Long Words (32 bits)

In addition, operations on other data types such as memory addresses, status word data, etc., are provided in the instruction set.

Most instructions can use any of the 14 addressing modes which are listed in Table 1-1. These addressing modes consist of six basic types:

- Register Direct
- Register Indirect
- Absolute
- Program Counter Relative
- Immediate
- Implied

Table 1-1. Addressing Modes

Addressing Modes	Syntax
Register Direct Addressing	
Data Register Direct	Dn
Address Register Direct	An
Absolute Data Addressing	
Absolute Short	xxx W
Absolute Long	xxx L
Program Counter Relative Addressing	
Relative with Offset	d_{16}(PC)
Relative with Index Offset	d_8(PC,Xn)
Register Indirect Addressing	
Register Indirect	(An)
Postincrement Register Indirect	(An) +
Predecrement Register Indirect	– (An)
Register Indirect with Offset	d_{16}(An)
Indexed Register Indirect with Offset	d_8(An,Xn)
Immediate Data Addressing	
Immediate	#xxx
Quick Immediate	#1–#8
Implied Addressing	
Implied Register	SR/USP/SP/PC

NOTES:

Dn = Data Register
An = Address Register
Xn = Address or Data Register used as Index Register
SR = Status Register
PC = Program Counter
SP = Stack Pointer
USP = User Stack Pointer
( ) = Contents of
d_8 = 8-Bit Offset (Displacement)
d_{16} = 16-Bit Offset (Displacement)
#xxx = Immediate Data

The register indirect addressing modes also have the capability to perform postincrementing, predecrementing, offsetting, and indexing. The program counter relative mode may be used in combination with indexing and offsetting for writing relocatable programs.

1.2 INSTRUCTION SET OVERVIEW

The TS68008 is completely code compatible with the TS68000. This means that programs developed for the TS68000 will run on the TS68008 and vice versa. This applies equally to either source code or object code.

The instruction set was designed to minimize the number of mnemonics remembered by the programmer. To further reduce the programmer's burden, the addressing modes are orthogonal.

The instruction set, shown in Table 1-2, forms a set of programming tools that include all processor functions to perform data movement, integer arithmetic, logical operations, shift and rotate operations, bit manipulation, BCD operations, and both program and system control. Some additional instructions are variations or subsets of these and appear in Table 1-3.

Table 1-2. Instruction Set

Mnemonic	Description
ABCD	Add Decimal With Extend
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
Bcc	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DBcc	Test Condition, Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive Or
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right

Mnemonic	Description
MOVE	Move
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical Or
PEA	Push Effective Address
RESET	Reset External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

Table 1-3. Variations of Instruction Types

Instruction Type	Variation	Description
ADD	ADD	Add
	ADDA	Add Address
	ADDQ	Add Quick
	ADDI	Add Immediate
	ADDX	Add with Extend
AND	AND	Logical And
	ANDI	And Immediate
	ANDI to CCR	And Immediate to Condition Codes
	ANDI to SR	And Immediate to Status Register
CMP	CMP	Compare
	CMPA	Compare Address
	CMPM	Compare Memory
	CMPI	Compare Immediate
EOR	EOR	Exclusive Or
	EORI	Exclusive Or Immediate
	EORI to CCR	Exclusive Or Immediate to Condition Codes
	EORI to SR	Exclusive Or Immediate to Status Register

Instruction Type	Variation	Description
MOVE	MOVE	Move
	MOVEA	Move Address
	MOVEC	Move Control Register
	MOVEM	Move Multiple Registers
	MOVEP	Move Peripheral Data
	MOVEQ	Move Quick
	MOVES	Move Alternate Address Space
	MOVE from SR	Move from Status Register
	MORE to SR	Move to Status Register
	MOVE from CCR	Move from Condition Codes
	MOVE to CCR	Move to Condition Codes
	MOVE USP	Move User Stack Pointer
NEG	NEG	Negate
	NEGX	Nexgate with Extend
OR	OR	Logical Or
	ORI	Or Immediate
	ORI to CCR	Or Immediate to Condition Codes
	ORI to SR	Or Immediate to Status Register
SUB	SUB	Subtract
	SUBA	Subtract Address
	SUBI	Subtract Immediate
	SUBQ	Subtract Quick
	SUBX	Subtract with Extend

SECTION 2
DATA ORGANIZATION AND ADDRESSING CAPABILITIES

This section describes the registers and data organization of the TS68008.

2.1 OPERAND SIZE

Operand sizes are defined as follows: a byte equals eight bits, a word equals 16 bits (two bytes), and a long word equals 32 bits (four bytes). The operand size for each instruction is either explicitly encoded in the instruction or implicitly defined by the instruction operation. Implicit instructions support some subset of all three sizes. When fetching instructions, the TS68008 always fetches pairs of bytes (words) thus guaranteeing compatibility with the TS68000.

2.2 DATA ORGANIZATION IN REGISTERS

The eight data registers support data operands of 1, 8, 16, or 32 bits. The seven address registers together with the stack pointers support address operands of 32 bits.

2.2.1 Data Registers

Each data register is 32 bits wide. Byte operands occupy the low order eight bits, word operands the low order 16 bits, and long word operands the entire 32 bits. The least significant bit is addressed as bit zero; the most significant bit is addressed as bit 31.

When a data register is used as either a source or destination operand, only the appropriate low order portion is changed; the remaining high order portion is neither used nor changed.

2.2.2 Address Registers

Each address register and the stack pointer is 32 bits wide and holds a full 32-bit address. Address registers do not support the byte sized operand. Therefore, when an address register is used as a source operand, either the low order word or the entire long word operand is used depending upon the operation size. When an address register is used as the destination operand, the entire register is affected regardless of the operation size. If the operation size is word, any other operands are sign extended to 32 bits before the operation is performed.

2.3 DATA ORGANIZATION IN MEMORY

The data types supported by the TS68008 are: bit data, integer data of 8, 16, or 32 bits, and 32-bit addresses. Figure 2-1 shows the organization of these data types in memory.

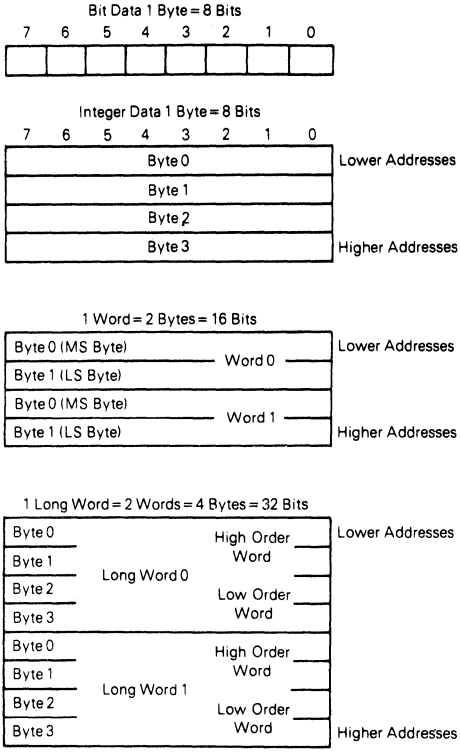


Figure 2-1. Memory Data Organization

2.4 ADDRESSING

Instructions for the TS68008 contain two kinds of information: the type of function to be performed, and the location of the operand(s) on which to perform that function. The methods used to locate (address) the operand(s) are explained in the following paragraphs.

Instructions specify an operand location in one of three ways:

Register Specification – the number of the register is given in the register field of the instruction.

Effective Address – use of the different effective address modes.

Implicit Reference – the definition of certain instructions implies the use of specific registers.

2.5 INSTRUCTION FORMAT

Instructions are from one to five words (two to ten bytes) in length as shown in Figure 2-2. Instructions always start on a word boundary thus guaranteeing compatibility with the TS68000. The length of the instruction and the operation to be performed is specified by the first word of the instruction which is called the operation word. The remaining words further specify the operands. These words are either immediate operands or extensions to the effective address mode specified in the operation word.

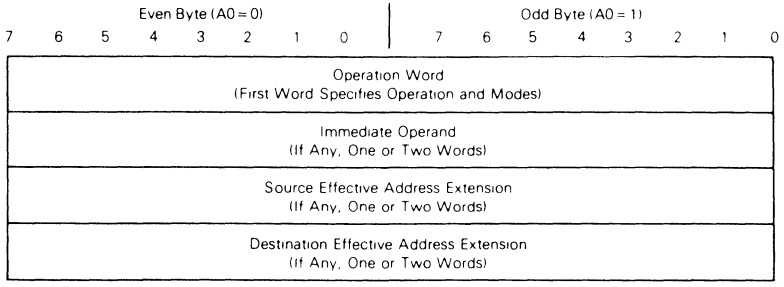


Figure 2-2. Instruction Operation Word General Format

2.6 PROGRAM/DATA REFERENCES

The TS68008 separates memory references into two classes: program references, and data references. Program references, as the name implies, are references to that section of memory containing the program being executed. Data references refer to that section of memory containing data. Operand reads are from the data space except in the case of the program counter relative addressing mode. All operand writes are to the data space. The function codes are used to indicate the address space being accessed during a bus cycle.

2.7 REGISTER SPECIFICATION

The register field within an instruction specifies the register to be used. Other fields within the instruction specify whether the register selected is an address or data register and how the register is to be used.

2.8 EFFECTIVE ADDRESS

Most instructions specify the location of an operand by using the effective address field in the operation word. For example, Figure 2-3 shows the general format of the single-effective-address instruction operation word. The effective address is composed of two 3-bit fields: the mode field, and the register field. The value in the mode field selects the different address modes. The register field contains the number of a register.

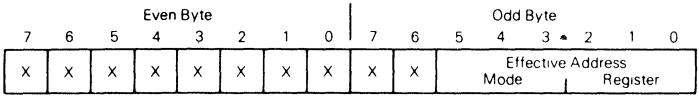


Figure 2-3. Single-Effective-Address Instruction Operation Word

The effective address field may require additional information to fully specify the operand. This additional information, called the effective address extension, is contained in the following word or words and is considered part of the instruction, as shown in Figure 2-2. The effective address modes are grouped into three categories: register direct, memory addressing, and special.

2.8.1 Register Direct Modes

These effective addressing modes specify that the operand is in one of sixteen multifunction registers.

2.8.1.1 DATA REGISTER DIRECT. The operand is in the data register specified by the effective address register field.

2.8.1.2 ADDRESS REGISTER DIRECT. The operand is in the address register specified by the effective address register field.

2.8.2 Memory Address Modes

These effective addressing modes specify that the operand is in memory and provide the specific address of the operand.

2.8.2.1 ADDRESS REGISTER INDIRECT. The address of the operand is in the address register specified by the register field. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.2.2 ADDRESS REGISTER INDIRECT WITH POSTINCREMENT. The address of the operand is in the address register specified by the register field. After the operand address is used, it is incremented by one, two, or four depending upon whether the size of the operand is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is incremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

2.8.2.3 ADDRESS REGISTER INDIRECT WITH PREDECREMENT. The address of the operand will be in the address register specified by the register field. Before the address register is used for operand access, it is decremented by one, two, or four depending upon whether the operand size is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is decremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

2.8.2.4 ADDRESS REGISTER INDIRECT WITH DISPLACEMENT. This address mode requires one word of extension. The address of the operand is the sum of the address in the address register and the sign-extended 16-bit displacement integer in the extension word. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.2.5 ADDRESS REGISTER INDIRECT WITH INDEX. This address mode requires one word of extension. The address of the operand is the sum of the address in the address register, the sign-extended displacement integer in the low order eight bits of the extension word, and the contents of the index register. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.3 Special Address Modes

The special address modes use the effective address register field to specify the special addressing mode instead of a register number.

2.8.3.1 ABSOLUTE SHORT ADDRESS. This address mode requires one word of extension. The address of the operand is the extension word. The 16-bit address is sign extended before it is used. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.3.2 ABSOLUTE LONG ADDRESS. This address mode requires two words of extension. The address of the operand is developed by the concatenation of the extension words. The high order part of the address is the first extension word; the low order part of the address is the second extension word. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

2.8.3.3 PROGRAM COUNTER WITH DISPLACEMENT. This address mode requires one word of extension. The address of the operand is the sum of the address in the program counter and the sign-extended 16-bit displacement integer in the extension word. The value in the program counter is the address of the extension word. The reference is classified as a program reference.

2.8.3.4 PROGRAM COUNTER WITH INDEX. This address mode requires one word of extension. This address is the sum of the address in the program counter, the sign-extended displacement integer in the lower eight bits of the extension word, and the contents of the index register. The value in the program counter is the address of the extension word. This reference is classified as a program reference.

2.8.3.5 IMMEDIATE DATA. This address mode requires either one or two words of extension depending on the size of the operation.

Byte Operation — operand is low order byte of extension word
Word Operation — operand is extension word
Long Word Operation — operand is in the two extension words, high order 16 bits are in the first extension word, low order 16 bits are in the second extension word.

2.8.3.6 IMPLICIT REFERENCE. Some instructions make implicit reference to the program counter (PC), the system stack pointer (SP), the supervisor stack pointer (SSP), the user stack pointer (USP), or the status register (SR). A selected set of instructions may reference the status register by means of the effective address field. These are:

ANDI to CCR	EORI to SR	MOVE to CCR
ANDI to SR	ORI to CCR	MOVE to SR
EORI to CCR	ORI to SR	MOVE from SR

2.9 EFFECTIVE ADDRESS ENCODING SUMMARY

Table 2-1 is a summary of the effective addressing modes discussed in the previous paragraphs.

Table 2-1. Effective Address Encoding Summary

Addressing Mode	Mode	Register
Data Register Direct	000	Register Number
Address Register Direct	001	Register Number
Address Register Indirect	010	Register Number
Address Register Indirect with Postincrement	011	Register Number
Address Register Indirect with Predecrement	100	Register Number
Address Register Indirect with Displacement	101	Register Number
Address Register Indirect with Index	110	Register Number
Absolute Short	111	000
Absolute Long	111	001
Program Counter with Displacement	111	010
Program Counter with Index	111	011
Immediate	111	100

2.10 SYSTEM STACK

The system stack is used implicitly by many instructions; user stacks and queues may be created and maintained through the addressing modes. Address register seven (A7) is the system stack pointer (SP). The system stack pointer is either the supervisor stack pointer (SSP) or the user stack pointer (USP), depending on the state of the S bit in the status register. If the S bit indicates supervisor state, SSP is the active system stack pointer and the USP is not used. If the S bit indicates user state, the USP is the active system stack pointer, and the SSP cannot be referenced. Each system stack fills from high memory to low memory.

SECTION 3
INSTRUCTION SET SUMMARY

This section contains an overview of the form and structure of the TS68008 instruction set. The instructions form a set of tools that include all the machine functions to perform the following operations:

Data Movement	Bit Manipulation
Integer Arithmetic	Binary Coded Decimal
Logical	Program Control
Shift and Rotate	System Control

The complete range of instruction capabilities combined with the flexible addressing modes described previously provide a very flexible base for program development.

3.1 DATA MOVEMENT OPERATIONS

The basic method of data acquisition (transfer and storage) is provided by the move (MOVE) instruction. The move instruction and the effective addressing modes allow both address and data manipulation. Data move instructions allow byte, word, and long word operands to be transferred from memory to memory, memory to register, register to memory, and register to register. Address move instructions allow word and long word operand transfers and ensure that only legal address manipulations are executed. In addition to the general move instruction there are several special data movement instructions: move multiple registers (MOVEM), move peripheral data (MOVEP), exchange registers (EXG), load effective address (LEA), push effective address (PEA), link stack (LINK), unlink stack (UNLK), and move quick (MOVEQ). Table 3-1 is a summary of the data movement operations.

Table 3-1. Data Movement Operations

Instruction	Operand Size	Operation
EXG	32	Rx ↔ Ry
LEA	32	EA → An
LINK	–	AN → – (SP) SP → An SP + displacement → SP
MOVE	8, 26, 32	(EA)s → (EA)d
MOVEM	16, 32	(EA) → An, Dn An, Dn → (EA)
MOVEP	16, 32	(EA) → Dn Dn → (EA)
MOVEQ	8	#xxx → Dn
PEA	32	EA → – (SP)
SWAP	32	Dn[31:16] ↔ Dn[15:0]
UNLK	–	An → SP (SP) + → An

NOTES:
s = source
d = destination
[ ] = bit number
– = indirect with predecrement
+ = indirect with postdecrement
# = immediate data

3.2 INTEGER ARITHMETIC OPERATIONS

The arithmetic operations include the four basic operations of add (ADD), subtract (SUB), multiply (MUL), and divide (DIV) as well as arithmetic compare (CMP), clear (CLR), and negate (NEG). The add and subtract instructions are available for both address and data operations, with data operations accepting all operand sizes. Address operations are limited to legal address size operands (16 or 32 bits). Data, address, and memory compare operations are also available. The clear and negate instructions may be used on all sizes of data operands.

The multiply and divide operations are available for signed and unsigned operands using word multiply to produce a long word product, and a long word dividend with word divisor to produce a word quotient with a word remainder.

Multiprecision and mixed size arithmetic can be accomplished using a set of extended instructions. These instructions are: add extended (ADDX), subtract extended (SUBX), sign extend (EXT), and negate binary with extend (NEGX).

A test operand (TST) instruction that will set the condition codes as a result of a compare of the operand with zero is also available. Test and set (TAS) is a synchronization instruction useful in multiprocessor systems. Table 3-2 is a summary of the integer arithmetic operations.

Table 3-2. Integer Arithmetic Operations

Instruction	Operand Size	Operation
ADD	8, 16, 32	Dn + (EA) → Dn (EA) + Dn → (EA) (EA) + #xxx → (EA)
	16, 32	An + (EA) → An
ADDX	8, 16, 32	Dx + Dy + X → Dx
	16, 32	−(Ax) + −(Ay) + X → (Ax)
CLR	8, 16, 32	0 → (EA)
CMP	8, 16, 32	Dn − (EA) (EA) − #xxxx (Ax)+ − (Ay)+
	16, 32	An − (EA)
DIVS	32 ÷ 16	Dn ÷ (EA) → Dn
DIVU	32 ÷ 16	Dn ÷ (EA) → Dn
EXT	8 → 16	$(Dn)_8 \rightarrow Dn_{16}$
	16 → 32	$(Dn)_{16} \rightarrow Dn_{32}$
MULS	16 × 16 → 32	Dn × (EA) → Dn
MULU	16 × 16 → 32	Dn × (EA) → Dn
NEG	8, 16, 32	0 − (EA) → (EA)
NEGX	8, 16, 32	0 − (EA) − X → (EA)
SUB	8, 16, 32	Dn − (EA) → Dn (EA) − Dn → (EA) (EA) − #xxx → (EA) An − (EA) → An
SUBX	8, 16, 32	Dx − Dy − X → Dx −(Ax) − −(Ay) − X → (Ax)
TAS	8	(EA) − 0, 1 → EA [7]
TST	8, 16, 32	(EA) − 0

NOTES:
[ ] = bit number
= immediate data
− = indirect with predecrement
+ = indirect with postdecrement

3.3 LOGICAL OPERATIONS

Logical operation instructions AND, OR, EOR, and NOT are available for all sizes of integer data operands. A similar set of immediate instructions (ANDI, ORI, and EORI) provide these logical operations with all sizes of immediate data. Table 3-3 is a summary of the logical operations.

Table 3-3. Logical Operations

Instruction	Operand Size	Operation
AND	8, 16, 32	Dn Λ (EA) → Dn (EA) Λ Dn → (EA) (EA) Λ #xxx → (EA)
OR	8, 16, 32	Dn V (EA) → Dn (EA) V Dn → (EA) (EA) V #xxx → (EA)
EOR	8, 16, 32	(EA) ⊕ Dy → (EA) (EA) ⊕ #xxx → (EA)
NOT	8, 16, 32	~(EA) → EA

NOTES:
= immediate data
~ = invert
Λ = logical AND
V = logical OR
⊕ = logical exclusive OR

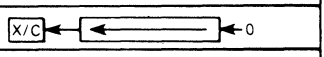
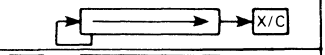
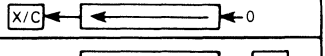
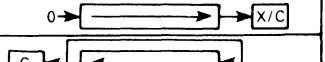
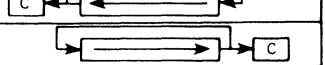
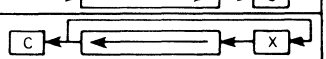
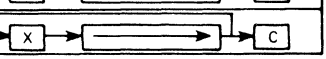
3.4 SHIFT AND ROTATE OPERATIONS

Shift operations in both directions are provided by the arithmetic instructions ASR and ASL and logical shift instructions LSR and LSL. The rotate instructions (with and without extend) available are ROXR, ROXL, ROR, and ROL. All shift and rotate operations can be performed in either registers or memory. Register shifts and rotates support all operand sizes and allow a shift count specified in a data register.

Memory shifts and rotates are for word operands and provide single-bit shifts or rotates.

Table 3-4 is a summary of the shift and rotate operations.

Table 3-4. Shift and Rotate Operations

Instruction	Operand Size	Operation
ASL	8, 16, 32	
ASR	8, 16, 32	
LSL	8, 16, 32	
LSR	8, 16, 32	
ROL	8, 16, 32	
ROR	8, 16, 32	
ROXL	8, 16, 32	
ROXR	8, 16, 32	

3.5 BIT MANIPULATION OPERATIONS

Bit manipulation operations are accomplished using the following instructions: bit test (BTST), bit test and set (BSET), bit test and clear (BCLR), and bit test and change (BCHG). Table 3-5 is a summary of the bit manipulation operations. (Z is bit 2 of the status register.)

Table 3-5. Bit Manipulation Operations

Instruction	Operand Size	Operation
BTST	8, 32	~bit of (EA) → Z
BSET	8, 32	~bit of (EA) → Z 1 → bit of EA
BCLR	8, 32	~bit of (EA) → Z 0 → bit of EA
BCHG	8, 32	~bit of (EA) → Z ~bit of (EA) → bit of EA

NOTE: ~ = invert

3.6 BINARY CODED DECIMAL OPERATIONS

Multiprecision arithmetic operations on binary coded decimal numbers are accomplished using the following instructions: add decimal with extend (ABCD), subtract decimal with extend (SBCD), and negate decimal with extend (NBCD). Table 3-6 is a summary of the binary coded decimal operations.

Table 3-6. Binary Coded Decimal Operations

Instruction	Operand Size	Operation
ABCD	8	$Dx_{10} + Dy_{10} + X \rightarrow Dx$ $-(Ax)_{10} + -(Ay)_{10} + X \rightarrow (Ax)$
SBCD	8	$Dx_{10} - Dy_{10} - X \rightarrow Dx$ $-(Ax)_{10} - -(Ay)_{10} - X \rightarrow (Ax)$
NBCD	8	$0 - (EA)_{10} - X \rightarrow (EA)$

3.7 PROGRAM CONTROL OPERATIONS

Program control operations are accomplished using a series of conditional and unconditional branch instructions, jump instructions, and return instructions. These instructions are summarized in Table 3-7.

The conditional instructions provide setting and branching for the following conditions:

CC – carry clear	GE – greater or equal	LS – low or same	PL – plus
CS – carry set	GT – greater than	LT – less than	T – always true
EQ – equal	HI – high	MI – minus	VC – no overflow
F – never true	LE – less or equal	NE – not equal	VS – overflow

Table 3-7. Program Control Operations

Instruction	Operation
Conditional	
B_{CC}	Branch Conditionally (14 conditions) 8- and 16-Bit Displacement
DB_{CC}	Test Condition, Decrement, and Branch 16-Bit Displacement
S_{CC}	Set Byte Conditionally (16 Conditions)
Unconditional	
BRA	Branch Always 8- and 16-Bit Displacement
BSR	Branch to Subroutine 8- and 16-Bit Displacement
JMP	Jump
JSR	Jump to Subroutine
Returns	
RTR	Return and Restore Condition Codes
RTS	Return from Subroutine

3.8 SYSTEM CONTROL OPERATIONS

System control operations are accomplished by using privileged instructions, trap generating instructions, and instructions that use or modify the status register. These instructions are summarized in Table 3-8.

Table 3-8. System Control Operations

Instruction	Operation
Privileged	
ANDI to SR	Logical AND to Status Register
EORI to SR	Logical EOR to Status Register
MOVE EA to SR	Load New Status Register
MOVE USP	Move User Stack Pointer
ORI to SR	Logical OR to Status Register
RESET	Reset External Devices
RTE	Return from Exception
STOP	Stop Program Execution
Trap Generating	
CHK	Check Data Register Against Upper Bounds
TRAP	Trap
TRAPV	Trap on Overflow
Status Register	
ANDI to CCR	Logical AND to Condition Codes
EORI to CCR	Logical EOR to Condition Codes
MOVE EA to CCR	Load New Condition Codes
MOVE SR to EA	Store Status Register
ORI to CCR	Logical OR to Condition Codes

SECTION 4
SIGNAL AND BUS OPERATION DESCRIPTION

This section contains a brief description of the input and output signals. A discussion of bus operation during the various machine cycles and operations is also given.

4.1 SIGNAL DESCRIPTION

The TS68008 is available in two package sizes (48-pin and 52-pin). The additional four pins of the 52-pin quad package allow for additional signals: A20, A21, $\overline{BGACK}$, and $\overline{IPL2}$.

Throughout this document, references to the address bus pins (A0-A19) and the interrupt priority level pins ($\overline{IPL0}$/$\overline{IPL2}$, $\overline{IPL1}$) refer to A0-A21 and $\overline{IPL0}$, $\overline{IPL1}$, and $\overline{IPL2}$ for the 52-pin version of the TS68008.

The input and output signals can be functionally organized into the groups shown in Figure 4-1(a) for the 48-pin version and in Figure 4-1(b) for the 52-pin version. The following paragraphs provide a brief description of the signals and a reference (if applicable) to other paragraphs that contain more information about the function being performed.

(a) 48-Pin Version

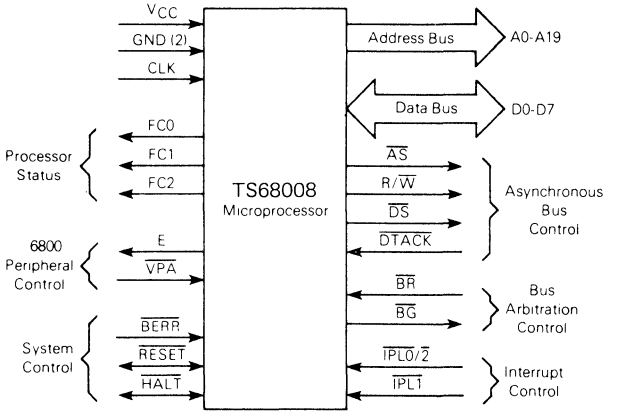


Figure 4-1. Input and Output Signals

(b) 52-Pin Version

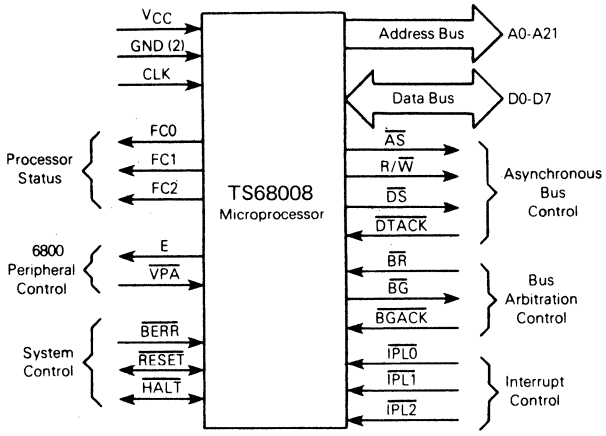


Figure 4-1. Input and Output Signals (Continued)

4.1.1 Address Bus (48-Pin: A0 through A19 52-Pin: A0 through A21)

This unidirectional three-state bus provides the address for bus operation during all cycles except interrupt acknowledge cycles. During interrupt acknowledge cycles, address lines A1, A2, and A3 provide information about what level interrupt is being serviced while address lines A0 and A4 through A19 (A21) are all driven high.

4.1.2 Data Bus (D0 through D7)

This 8-bit, bidirectional, three-state bus is the general purpose data path. During an interrupt acknowledge cycle, the external device supplies the vector number on data lines D0-D7.

4.1.3 Asynchronous Bus Control

Asynchronous data transfers are handled using the following control signals: address strobe, read/write, data strobe, and data transfer acknowledge. These signals are explained in the following paragraphs.

4.1.3.1 ADDRESS STROBE ($\overline{AS}$). This three-state signal indicates that there is a valid address on the address bus. It is also used to "lock" the bus during the read-modify-write cycle used by the test and set (TAS) instruction.

4.1.3.2 READ/WRITE (R/$\overline{W}$). This three-state signal defines the data bus transfer as a read or write cycle. The R/$\overline{W}$ signal also works in conjunction with the data strobe as explained in the following paragraph.

4.1.3.3 DATA STROBE ($\overline{DS}$). This three-state signal controls the flow of data on the data bus as shown in Table 4-1. When the R/$\overline{W}$ line is high, the processor will read from the data bus as indicated. When the R/$\overline{W}$ line is low, the processor will write to the data bus as shown.

Table 4-1. Data Strobe Control of Data Bus

$\overline{DS}$	R/$\overline{W}$	D0-D7
1	–	No Valid Data
0	1	Valid Data Bits 0-7 (Read Cycle)
0	0	Valid Data Bits 0-7 (Write Cycle)

4.1.3.4 DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$). This input indicates that the data transfer is completed. When the processor recognizes $\overline{DTACK}$ during a read cycle, data is latched and the bus cycle is terminated. When $\overline{DTACK}$ is recognized during a write cycle, the bus cycle is terminated. (Refer to **4.4 ASYNCHRONOUS VERSUS SYNCHRONOUS OPERATION**.)

4.1.4 Bus Arbitration Control

The 48-pin TS68008 contains a simple two-wire arbitration circuit and the 52-pin TS68008 contains the full three-wire TS68000 bus arbitration control. Both versions are designed to work with daisy-chained networks, priority encoded networks, or a combination of these techniques. This circuit is used in determining which device will be the bus master device.

4.1.4.1 BUS REQUEST ($\overline{BR}$). This input is wire ORed with all other devices that could be bus masters. This device indicates to the processor that some other device desires to become the bus master. Bus requests may be issued at any time in a cycle or even if no cycle is being performed.

4.1.4.2 BUS GRANT ($\overline{BG}$). This output indicates to all other potential bus master devices that the processor will release bus control at the end of the current bus cycle.

4.1.4.3 BUS GRANT ACKNOWLEDGE ($\overline{BGACK}$). This input, available on the 52-pin version only, indicates that some other device has become the bus master. This signal should not be asserted until the following four conditions are met:

1. a bus grant has been received,
2. address strobe is inactive which indicates that the microprocessor is not using the bus,
3. data transfer acknowledge is inactive which indicates that neither memory nor peripherals are using the bus, and
4. bus grant acknowledge is inactive which indicates that no other device is still claiming bus mastership.

NOTES

1) There is a two-clock interval straddling the transition of $\overline{AS}$ from the inactive state to the active state during which $\overline{BG}$ cannot be issued.
2) If an existing TS68000 system is retrofitted to use the TS68008, 48-pin version (using $\overline{BR}$ and $\overline{BG}$ only), the existing $\overline{BR}$ and $\overline{BGACK}$ signals should be ANDed and the resultant signal connected to the TS68008's $\overline{BR}$.

4.1.5 Interrupt Control (48-Pin: $\overline{IPL0}/\overline{IPL2}$, $\overline{IPL1}$ 52-Pin: $\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$)

These input pins indicate the encoded priority level of the device requesting an interrupt. The TS68000 and the 52-pin TS68008 MPUs use three pins to encode a range of 0-7 but, for the 48-pin TS68008 only two pins are available. By connecting the $\overline{IPL0}/\overline{IPL2}$ pin to both the $\overline{IPL0}$ and $\overline{IPL2}$ inputs internally, the 48-pin encodes values of 0, 2, 5, and 7. Level zero is used to indicate that there are no interrupts pending and level seven is a non-maskable edge-triggered interrupt. Except for level seven, the requesting level must be greater than the interrupt mask level contained in the processor status register before the processor will acknowledge the request.

The level presented to these inputs is continually monitored to allow for the case of a requesting level that is less than or equal to the processor status register level to be followed by a request that is greater than the processor status register level. A satisfactory interrupt condition must exist for two successive clocks before triggering an internal interrupt request. An interrupt acknowledge sequence is indicated by the function codes.

4.1.6 System Control

The system control inputs are used to either reset or halt the processor and to indicate to the processor that bus errors have occurred. The three system control signals are explained in the following paragraphs.

4.1.6.1 BUS ERROR ($\overline{BERR}$). This input informs the processor that there is a problem with the cycle currently being executed. Problems may be a result of:

1. nonresponding devices,
2. interrupt vector number acquisition failure,
3. illegal access request as determined by a memory management unit, or
4. various other application dependent errors.

The bus error signal interacts with the halt signal to determine if the current bus cycle should be re-executed or if exception processing should be performed. Refer to **4.2.3 Bus Error and Halt Operation** for a detailed description of the interaction which is summarized below.

$\overline{BERR}$	$\overline{HALT}$	Resulting Operation
High	High	Normal operation
High	Low	Single bus cycle operation
Low	High	Bus error – exception processing
Low	Low	Bus error – re-run current cycle

4.1.6.2 RESET ($\overline{RESET}$). This bidirectional signal line acts to reset (start a system initialization sequence) the processor in response to an external $\overline{RESET}$ signal. An internally generated reset (result of a reset instruction) causes all external devices to be reset and the internal state of the processor is not affected. A total system reset (processor and external devices) is the result of external $\overline{HALT}$ and $\overline{RESET}$ signals applied at the same time. Refer to **4.2.4 Reset Operation** for further information.

4.1.6.3 HALT ($\overline{HALT}$). When this bidirectional line is driven by an external device, it will cause the processor to stop at the completion of the current bus cycle. When the processor has been halted

using this input, all control signals are inactive and all three-state lines are put in their high-impedance state. Refer to **4.2.3 Bus Error and Halt Operation** for additional information about the interaction between the halt and bus error signals.

When the processor has stopped executing instructions, such as in a double bus fault condition, the halt line is driven by the processor to indicate to external devices that the processor has stopped.

4.1.7 6800 Peripheral Control

These control signals are used to allow the interfacing of synchronous 6800 peripheral devices with the asynchronous TS68008. These signals are explained in the following paragraphs.

The TS68008 does not supply a valid memory address ($\overline{VMA}$) signal like that of the TS68000. The $\overline{VMA}$ signal indicates to the 6800 peripheral devices that there is a valid address on the address bus and that the processor is synchronized to the enable clock. This signal can be produced by a TTL circuit (see a sample circuit in Figure 4-2). The $\overline{VMA}$ signal, in this circuit, only responds to a valid peripheral address (VPA) input which indicates that the peripheral is an 6800 Family device. Timing for this circuit is shown in Figure 6-2.

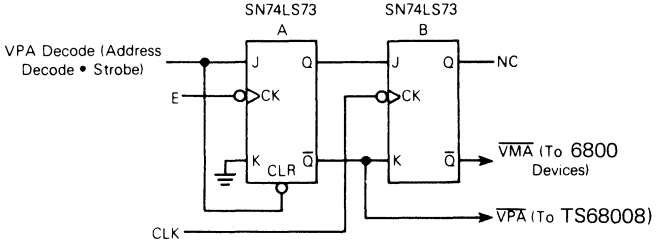


Figure 4-2. External $\overline{VMA}$ Generation

The VPA decode shown in Figure 4-2 is an active high decode indicating that address strobe ($\overline{AS}$) has been asserted and the address bus is addressing an 6800 peripheral. The $\overline{VPA}$ output of the circuit is used to indicate to the TS68008 that the data transfer should be synchronized with the enable (E) signal.

4.1.7.1 ENABLE (E). This signal is the standard enable signal common to all 6800 type peripheral devices. The period for this output is 10 TS68008 clock periods (six clocks low, four clocks high).

4.1.7.2 VALID PERIPHERAL ADDRESS ($\overline{VPA}$). This input indicates that the device or region addressed is a 6800 Family device and that data transfer should be synchronized with the enable (E) signal. This input also indicates that the processor should use automatic vectoring for an interrupt. Refer to **6.0 INTERFACE WITH 6800 PERIPHERALS.**

4.1.8 Processor Status (FC0, FC1, FC2)

These function code outputs indicate the state (user or supervisor) and the cycle type currently being executed, as shown in Table 4-2. The information indicated by the function code outputs is valid whenever address strobe ($\overline{AS}$) is active.

Table 4-2. Function Code Outputs

Function Code Output			
FC2	FC1	FC0	Cycle Type
Low	Low	Low	(Undefined, Reserved)
Low	Low	High	User Data
Low	High	Low	User Program
Low	High	High	(Undefined, Reserved)
High	Low	Low	(Undefined, Reserved)
High	Low	High	Supervisor Data
High	High	Low	Supervisor Program
High	High	High	Interrupt Acknowledge

4.1.9 Clock (CLK)

The clock input is a TTL-compatible signal that is internally buffered for development of the internal clocks needed by the processor. The clock input shall be a constant frequency.

4.1.10 V_{CC} and GND

Power is supplied to the processor using these two signals. V_{CC} is power and GND is the ground connection.

4.1.11 Signal Summary

Table 4-3 is a summary of all the signals discussed in the previous paragraphs.

Table 4-3. Signal Summary

Signal Name	Mnemnoic	Input/Output	Active State	Hi-Z	
				on $\overline{HALT}$	on $\overline{BGACK}$
Address Bus	A0-A19	Output	High	Yes	Yes
Data Bus	D0-D7	Input/Output	High	Yes	Yes
Address Strobe	$\overline{AS}$	Output	Low	No	Yes
Read/Write	R/$\overline{W}$	Output	Read-High Write-Low	No No	Yes Yes
Data Strobes	$\overline{DS}$	Output	Low	No	Yes
Data Transfer Acknowledge	$\overline{DTACK}$	Input	Low	No	No
Bus Request	$\overline{BR}$	Input	Low	No	No
Bus Grant	$\overline{BG}$	Output	Low	No	No
Bus Grant Acknowledge**	$\overline{BGACK}$	Input	Low	No	No
Interrupt Priority Level	$\overline{IPLx}$	Input	Low	No	No
Bus Error	$\overline{BERR}$	Input	Low	No	No
Reset	$\overline{RESET}$	Input/Output	Low	No*	No*
Halt	$\overline{HALT}$	Input/Output	Low	No*	No*
Enable	E	Output	High	No	No
Valid Peripheral Address	$\overline{VPA}$	Input	Low	No	No
Function Code Output	FC0, FC1, FC2	Output	High	No	Yes
Clock	CLK	Input	High	No	No
Power Input	V_{CC}	Input	–	–	–
Ground	GND	Input	–	–	–

*Open Drain

**52-Pin Version Only

4.2 BUS OPERATION

The following paragraphs explain control signal and bus operation during data transfer operations, bus arbitration, bus error and halt conditions, and reset operation.

4.2.1 Data Transfer Operations

Transfer of data between devices involves the following leads:

- Address bus A0 through A19
- Data bus D0 through D7
- Control signals

The address and data buses are separate non-multiplexed parallel buses. Data transfer is accomplished with an asynchronous bus structure that uses handshakes to ensure the correct movement of data. In all cycles, the bus master assumes responsibility for deskewing all signals it issues at both the start and end of a cycle. In addition, the bus master is responsible for deskewing the acknowledge and data signals from the slave device.

The following paragraphs explain the read, write, and read-modify-write cycles. The indivisible read-modify-write cycle is the method used by the TS68008 for interlocked multiprocessor communications.

NOTE

The terms **assertion** and **negation** will be used extensively. This is done to avoid confusion when dealing with a mixture of "active-low" and "active-high" signals. The term assert or assertion is used to indicate that a signal is active or true independent of whether that voltage is low or high. The term negate or negation is used to indicate that a signal is inactive or false.

4.2.1.1 READ CYCLE. During a read cycle, the processor receives data from the memory or a peripheral device. The processor reads bytes of data in all cases. If the instruction specifies a word (or double word) operation, the processor reads both bytes. When the instruction specifies byte operation, the processor uses A0 to determine which byte to read and then issues data strobe.

A word read cycle flowchart is given in Figure 4-3. A byte read cycle flowchart is given in Figure 4-4. Read cycle timing is given in Figure 4-5. Figure 4-6 details words and byte read cycle operations.

4.2.1.2 WRITE CYCLE. During a write cycle, the processor sends data to either the memory or a peripheral device. The processor writes bytes of data in all cases. If the instruction specifies a word operation, the processor writes both bytes. When the instruction specifies a byte operation, the processor uses A0 to determine which byte to write and then issues the data strobe. A word write cycle flowchart is given in Figure 4-7. A byte write cycle flowchart is given in Figure 4-8. Write cycle timing is given in Figure 4-5. Figure 4-9 details word and byte write cycle operation.

4.2.1.3 READ-MODIFY-WRITE CYCLE. The read-modify-write cycle performs a byte read, modifies the data in the arithmetic-logic unit, and writes the data back to the same address. In the TS68008, this cycle is indivisible in that the address strobe is asserted throughout the entire cycle. The test and set (TAS) instruction uses this cycle to provide meaningful communication between processors in a multiple processor environment. This instruction is the only instruction that uses the read-modify-write cycle and since the test and set instruction only operates on bytes, all read-modify-write cycles are byte operations. A read-modify-write cycle flowchart is given in Figure 4-10 and a timing diagram is given in Figure 4-11.

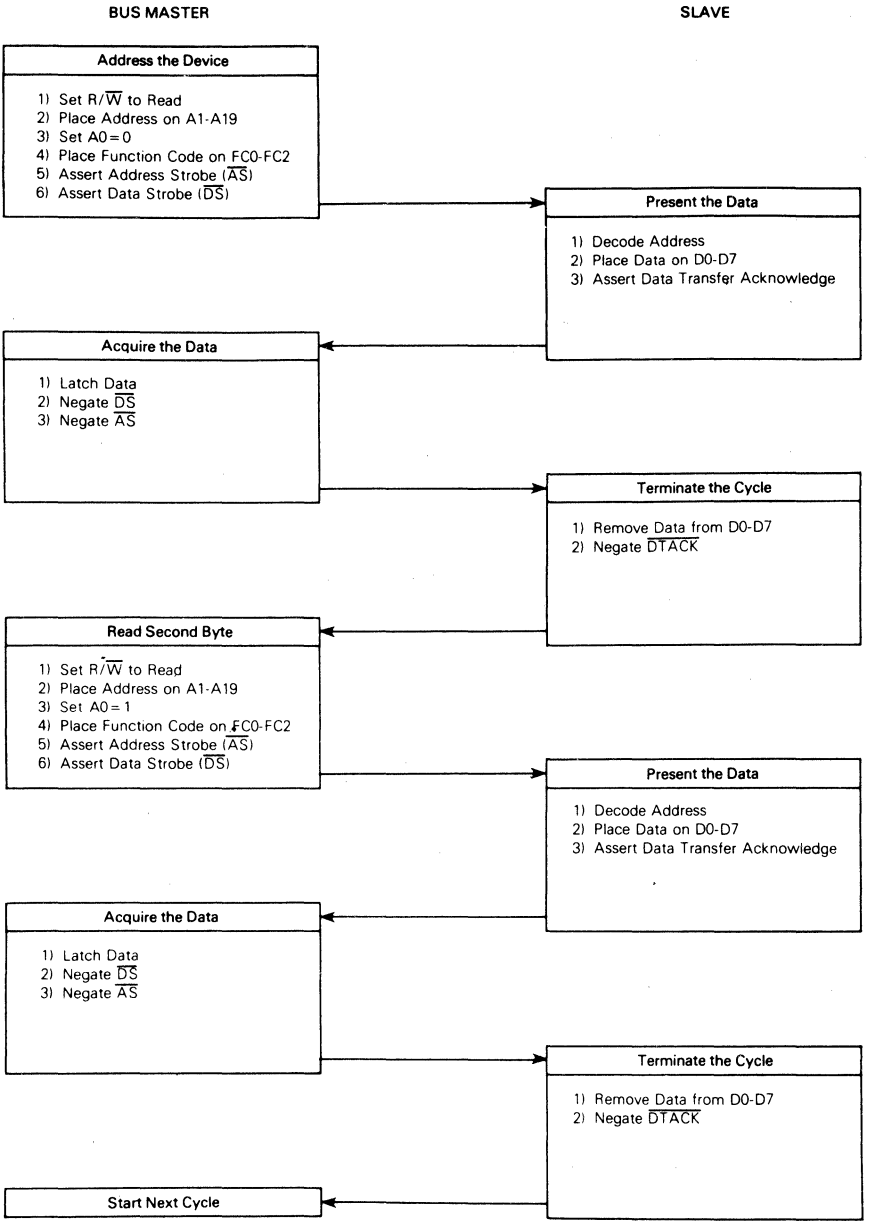


Figure 4-3. Word Read Cycle Flowchart

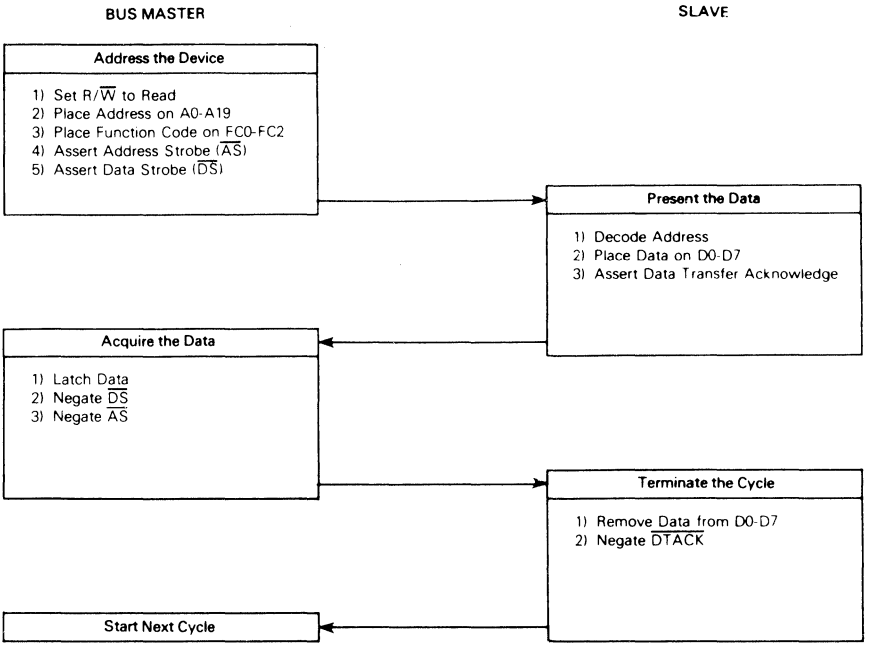


Figure 4-4. Byte Read Cycle Flowchart

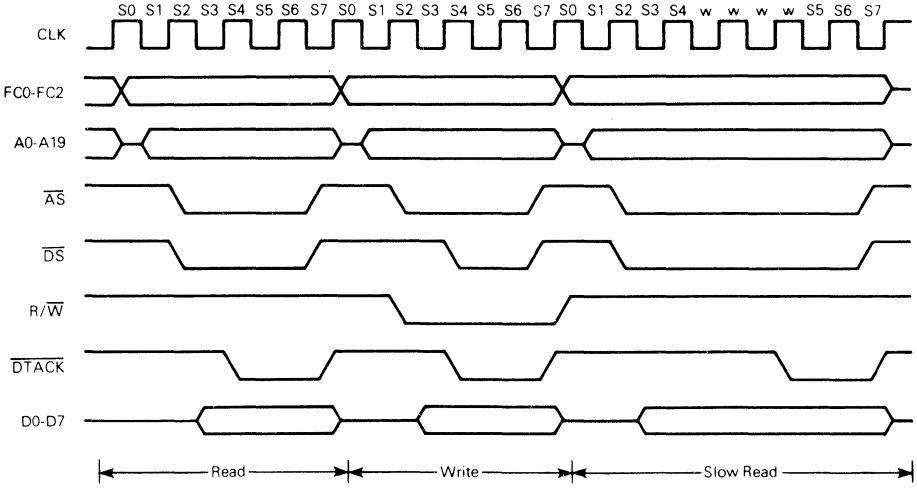


Figure 4-5. Read and Write Cycle Timing Diagram

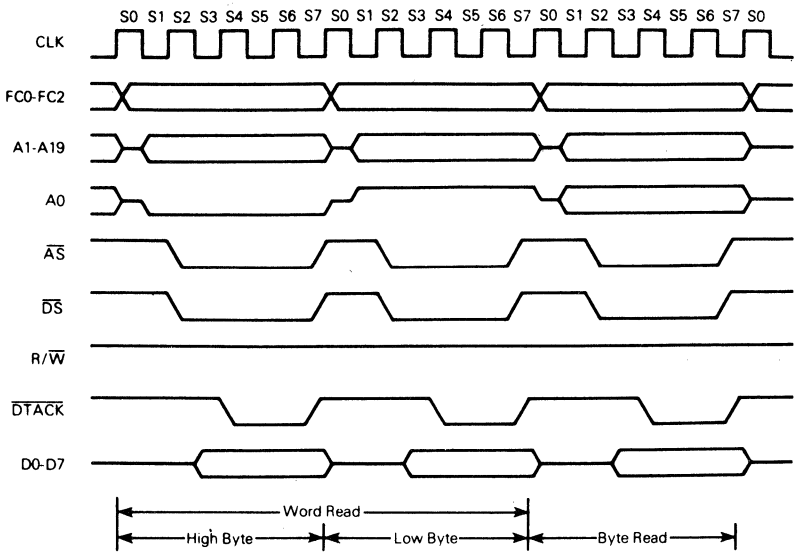


Figure 4-6. Word and Byte Read Cycle Timing

4.2.2 Bus Arbitration

Bus arbitration on the 52-pin version of the TS68008 is identical to that on the TS68000.

Bus arbitration on the 48-pin version of the TS68008 has been modified from that on the TS68000. It is controlled by the same finite state machine as on the TS68000, but because the $\overline{\text{BGACK}}$ input signal is not bonded out to a pin and is, instead, permanently negated internally, the bus arbitration becomes a two-wire handshake circuit. Therefore, in reading the following paragraphs for a description of bus arbitration on the 48-pin version of the TS68008, the $\overline{\text{BGACK}}$ signal should be considered permanently negated.

Bus arbitration is a technique used by master-type devices to request, be granted, and acknowledge bus mastership. In its simplest form, it consists of the following:

1. asserting a bus mastership request,
2. receiving a grant that the bus is available at the end of the current cycle, and
3. on the 52-pin version of the TS68008 only, acknowledging that mastership has been assumed.

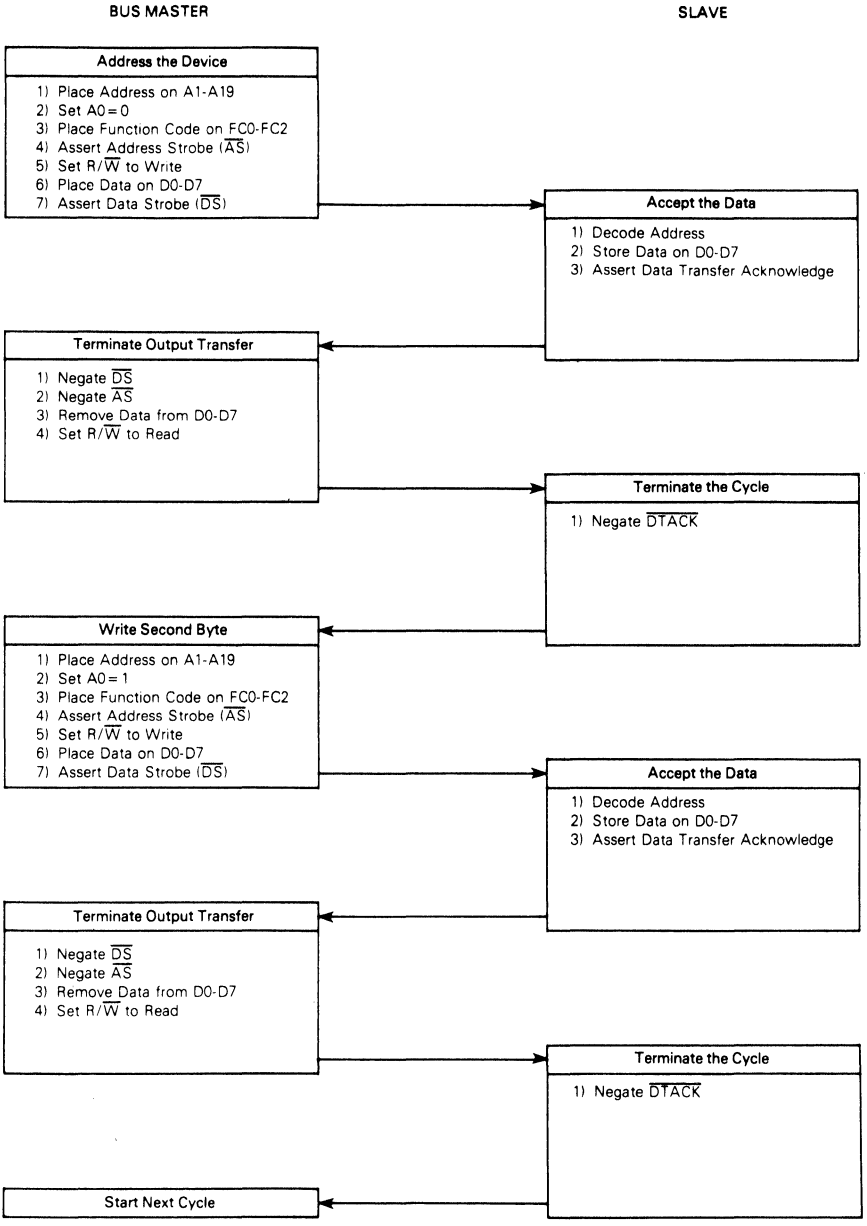


Figure 4-7. Word Write Cycle Flowchart

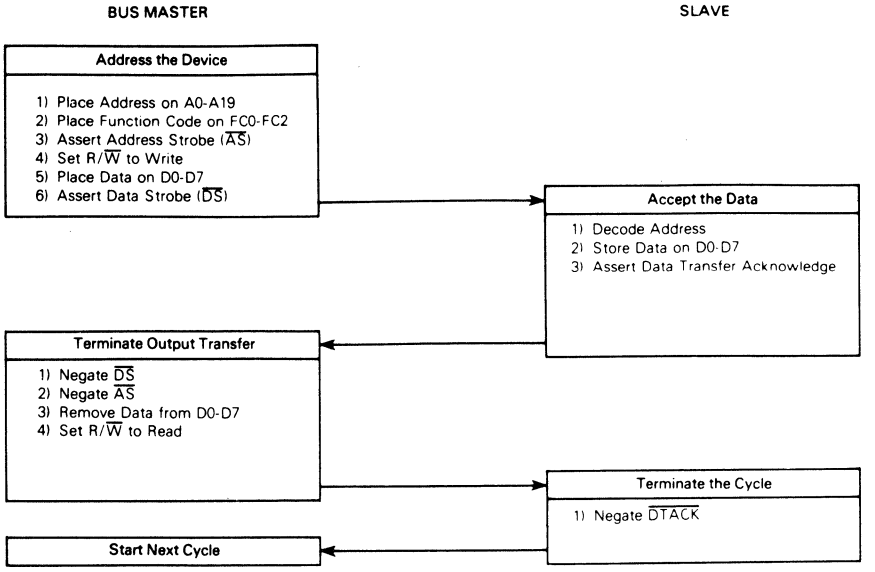


Figure 4-8. Byte Write Cycle Flowchart

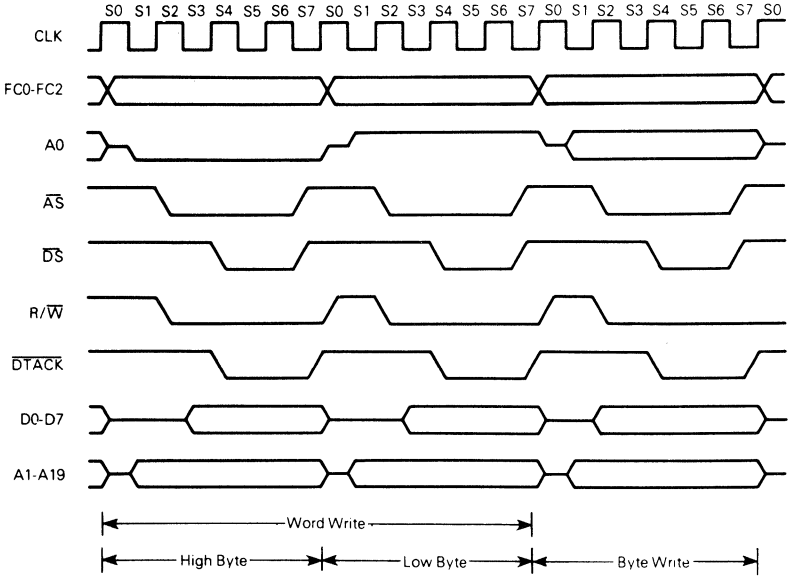


Figure 4-9. Word and Byte Write Cycle Timing

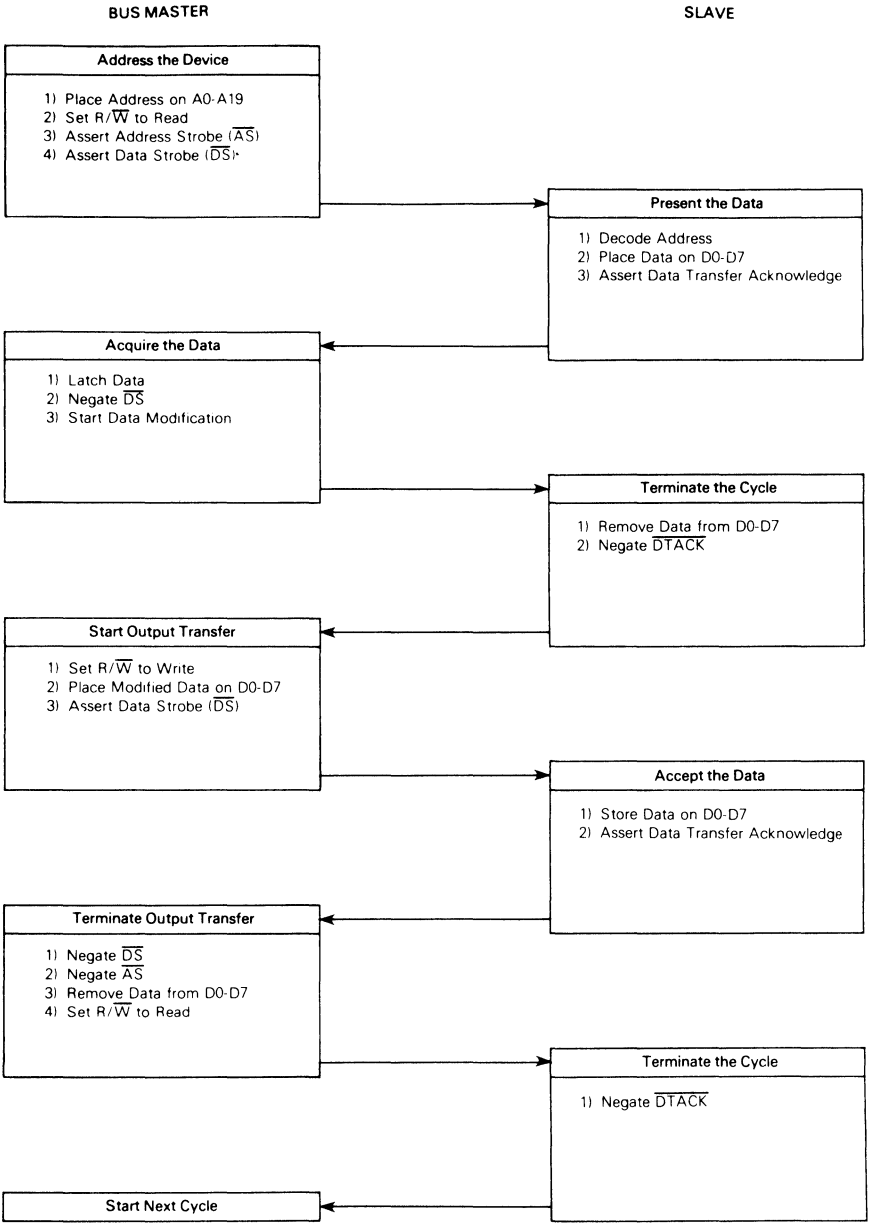


Figure 4-10. Read-Modify-Write Cycle Flowchart

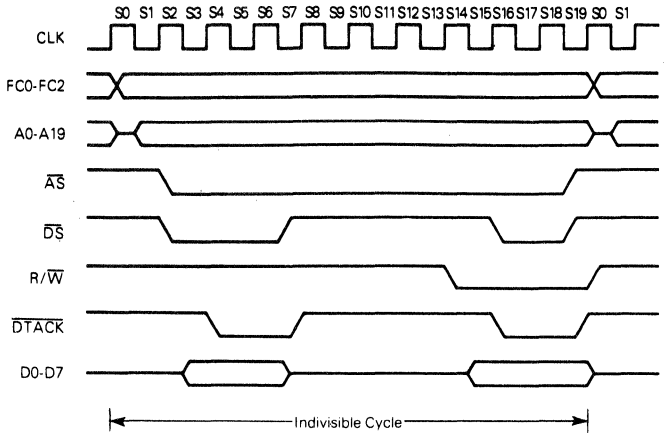


Figure 4-11. Read-Modify-Write Cycle Timing

Flowcharts showing the detail involved in a request from a single device are illustrated in Figure 4-12 for the 48-pin version and Figure 4-13 for the 52-pin version. Timing diagrams for the same operation are given in Figure 4-14 and Figure 4-15. This technique allows processing of bus requests during data transfer cycles.

The timing diagram shows that the bus request is negated at the time that an acknowledge is asserted. This type of operation would be true for a system consisting of the processor and one device capable of bus mastership. In systems having a number of devices capable of bus mastership, the bus request line from each device is wire ORed to the processor. In this system, it is easy to see that there could be more than one bus request being made. The timing diagram shows that the bus grant signal is negated a few clock cycles after the transition of the acknowledge ($\overline{BGACK}$) signal.

However, if the bus requests are still pending, the processor will assert another bus grant within a few clock cycles after it was negated. This additional assertion of bus grant allows external arbitration circuitry to select the next bus master before the current bus master has completed its requirements. The following paragraphs provide additional information about the three steps in the arbitration process.

4.2.2.1 REQUESTING THE BUS. External devices capable of becoming bus masters request the bus by asserting the bus request ($\overline{BR}$) signal. This is a wire-ORed signal (although it need not be constructed from open-collector devices) that indicates to the processor that some external device requires control of the external bus. The processor is effectively at a lower bus priority level than the external device and will relinquish the bus after it has completed the last bus cycle it has started.

On the 52-pin version, when no acknowledge is received before the bus request signal goes inactive, the processor will continue processing when it detects that the bus request is inactive. This allows ordinary processing to continue if the arbitration circuitry responded to noise inadvertently.

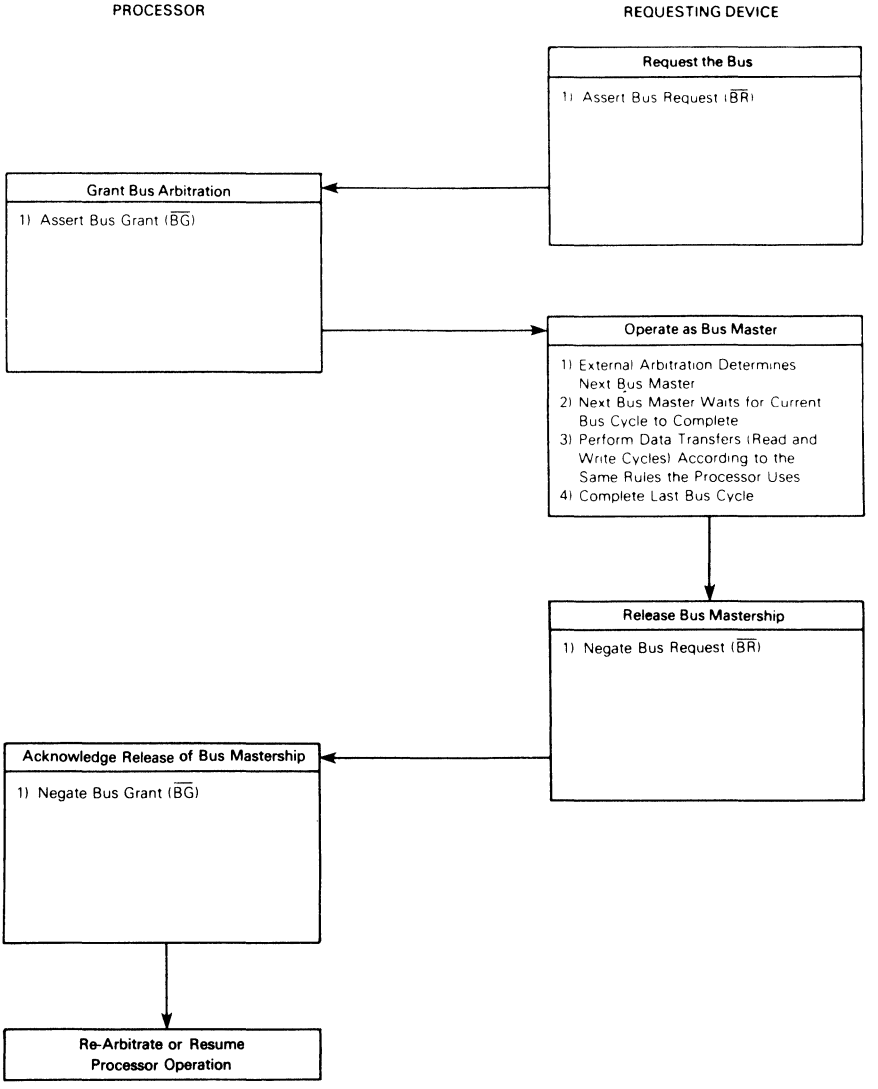


Figure 4-12. Bus Arbitration Cycle Flowchart for the 48-Pin Version

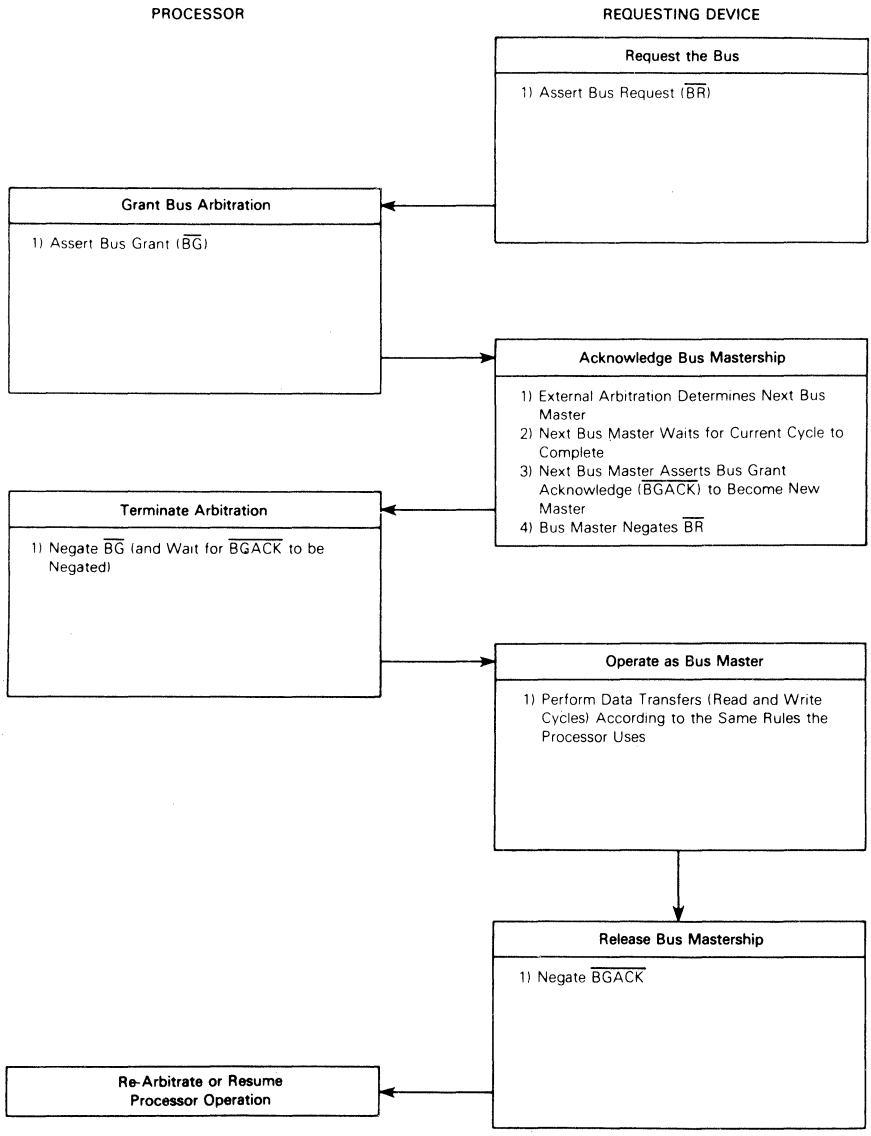


Figure 4-13. Bus Arbitration Cycle Flowchart for the 52-Pin Version

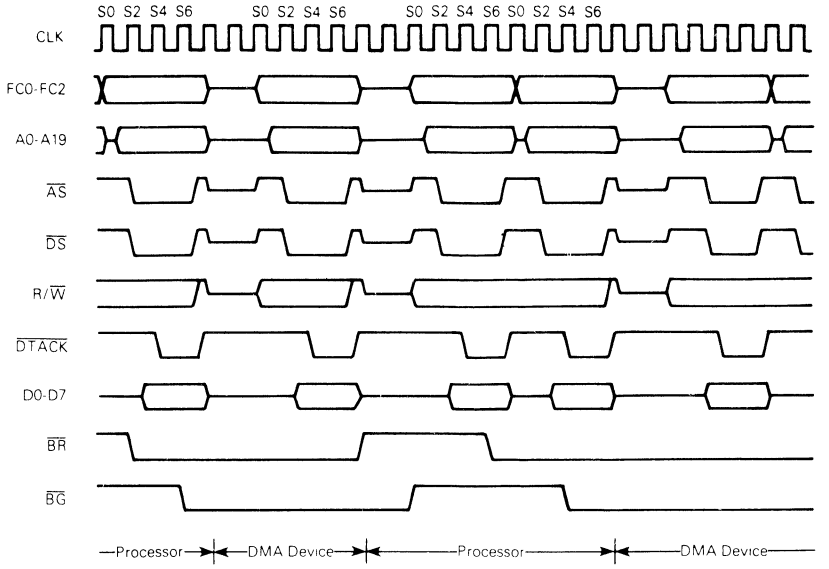


Figure 4-14. Bus Arbitration Timing for the 48-Pin Version

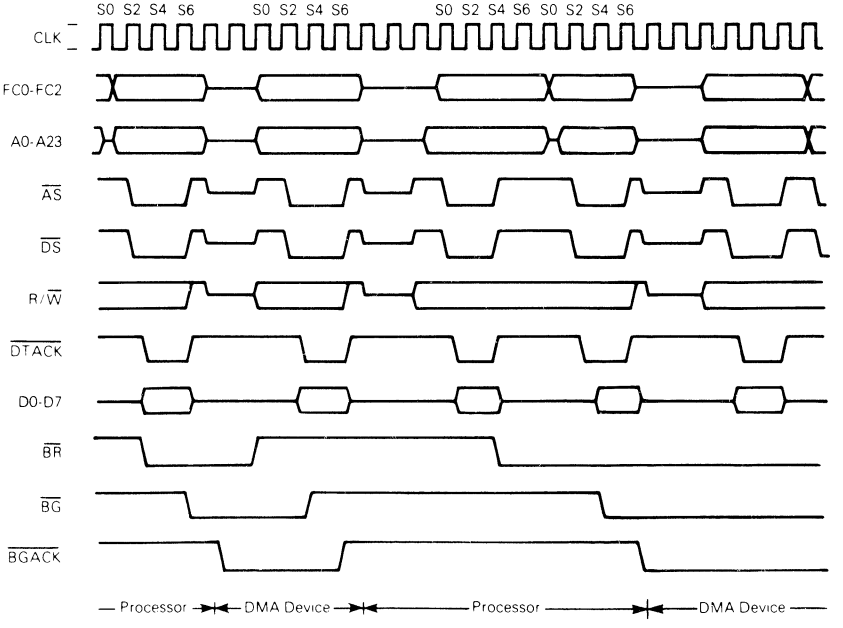


Figure 4-15. Bus Arbitration Timing for the 52-Pin Version

4.2.2.2 RECEIVING THE BUS GRANT. The processor asserts bus grant ($\overline{BG}$) as soon as possible. Normally this is immediately after internal synchronization. The only exception to this occurs when the processor has made an internal decision to execute the next bus cycle but has not progressed far enough into the cycle to have asserted the address strobe ($\overline{AS}$) signal. In this case, bus grant will be delayed until $\overline{AS}$ is asserted to indicate to external devices that a bus cycle is being executed.

The bus grant signal may be routed through a daisy-chained network or through a specific priority-encoded network. The processor is not affected by the external method of arbitration as long as the protocol is obeyed.

4.2.2.3 ACKNOWLEDGEMENT OF MASTERSHIP (52-PIN VERSION OF TS68008 ONLY). Upon receiving a bus grant, the requesting device waits until address strobe, data transfer acknowledge, and bus grant acknowledge are negated before issuing its own $\overline{BGACK}$. The negation of the address strobe indicates that the previous master has completed its cycle; the negation of bus grant acknowledge indicates that the previous master has released the bus. (While address strobe is asserted, no device is allowed to "break into" a cycle.) The negation of data transfer acknowledge indicates the previous slave has terminated its connection to the previous master. Note that in some applications data transfer acknowledge might not enter into this function. General purpose devices would then be connected such that they were only dependent on address strobe. When bus grant acknowledge is issued, the device is a bus master until it negates bus grant acknowledge. Bus grant acknowledge should not be negated until after the bus cycle(s) is (are) completed. Bus mastership is terminated at the negation of bus grant acknowledge.

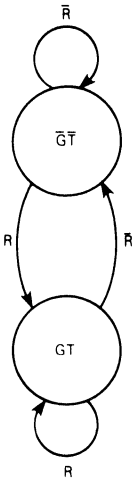
The bus request from the granted device should be dropped after bus grant acknowledge is asserted. If a bus request is still pending, another bus grant will be asserted within a few clocks of the negation of the bus grant. Refer to **4.2.3 Bus Arbitration Control Unit.** Note that the processor does not perform any external bus cycles before it re-asserts bus grant.

4.2.3 Bus Arbitration Control

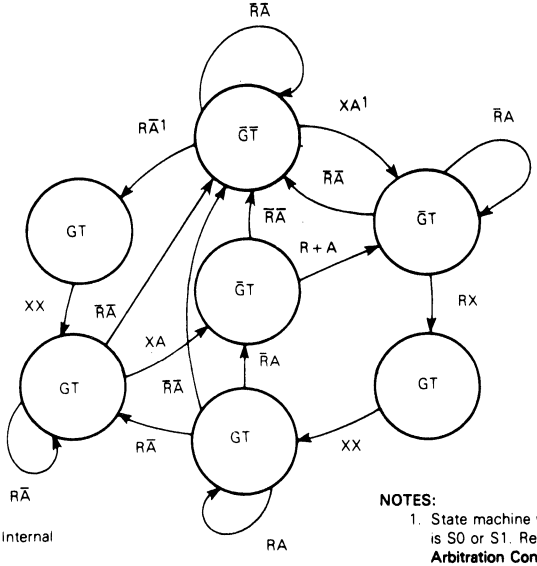
The bus arbitration control unit in the TS68008 is implemented with a finite state machine. A state diagram of this machine is shown in Figure 4-16 for both pin versions of the TS68008. All asynchronous signals to the TS68008 are synchronized before being used internally. This synchronization is accomplished in a maximum of one cycle of the system clock, assuming that the asynchronous input setup time (#47) has been met (see Figure 4-17). The input signal is sampled on the falling edge of the clock and is valid internally after the next falling edge.

As shown in Figure 4-16, input signals labeled R and A are internally synchronized on the bus request and bus grant acknowledge pins respectively. The bus grant output is labeled G and the internal three-state control signal T. If T is true, the address, data, and control buses are placed in a high-impedance state when $\overline{AS}$ is negated. All signals are shown in positive logic (active high) regardless of their true active voltage level. State changes (valid outputs) occur on the next rising edge after the internal signal is valid.

(a) State Diagram for the 48-Pin Version of TS68008



(b) State Diagram for the 52-Pin Version of **TS68008**



R = Bus Request Internal
A = Bus Grant Acknowledge Internal
G = Bus Grant
T = Three-State Control to Bus Control Logic[2]
X = Don't Care

NOTES:
1. State machine will not change if the bus is S0 or S1. Refer to **4.2.3 Bus Arbitration Contol.**
2. The address bus will be placed in the high-impedance state if T is asserted and $\overline{AS}$ is negated.

Figure 4-16. **TS68008** Bus Arbitration Unit State Diagram

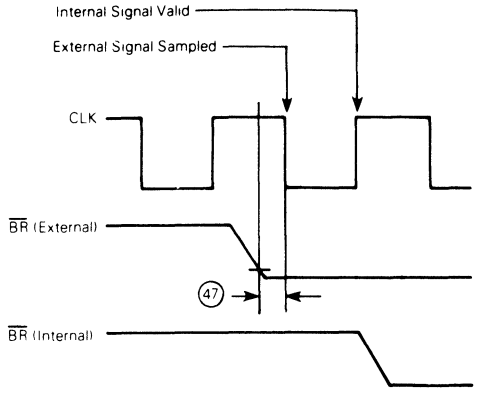


Figure 4-17. Timing Relationship of External Asynchronous Inputs to Internal Signals

A timing diagram of the bus arbitration sequence during a processor bus cycle is shown in Figure 4-18. The bus arbitration sequence while the bus is inactive (i.e., executing internal operations such as a multiply instruction) is shown in Figure 4-19.

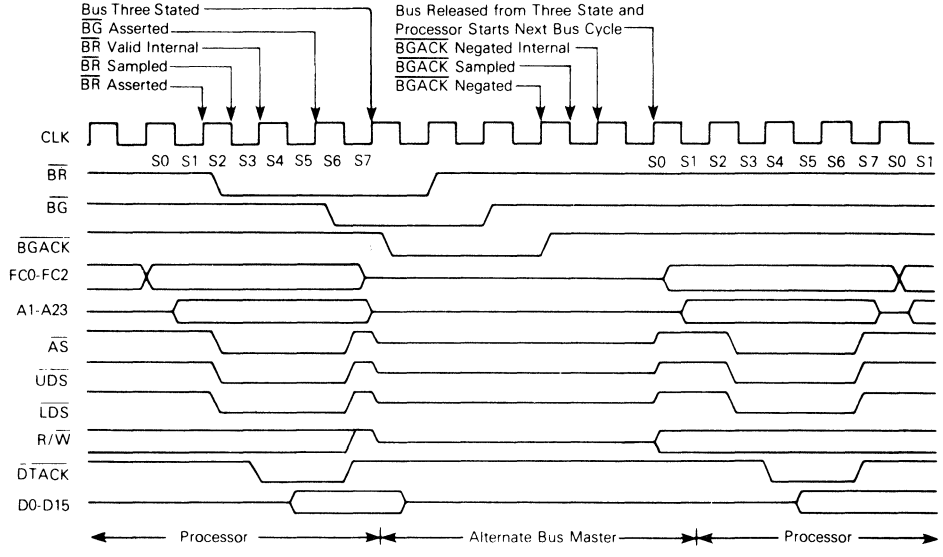


Figure 4-18. Bus Arbitration Timing Diagram—Processor Active

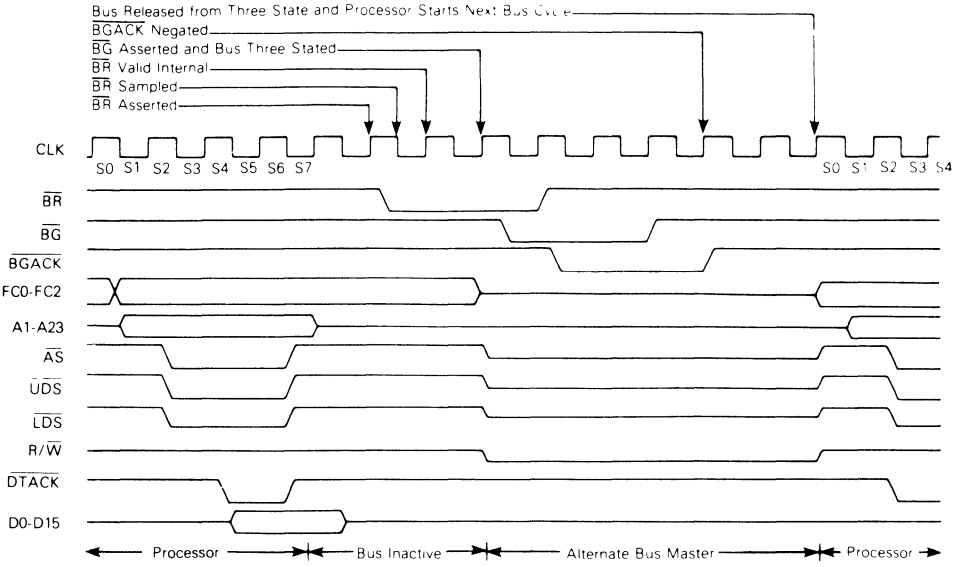


Figure 4-19. Bus Arbitration Timing Diagram—Bus Inactive

If a bus request is made at a time when the MPU has already begun as bus cycle but $\overline{AS}$ has not been asserted (bus state S0), $\overline{BG}$ will not be asserted on the next rising edge. Instead, $\overline{BG}$ will be delayed until the second rising edge following its internal assertion. This sequence is shown in Figure 4-20.

4.2.4 Bus Error and Halt Operation

In a bus architecture that requires a handshake from an external device, the possibility exists that the handshake might not occur. Since different systems will require a different maximum response time, a bus error input is provided. External circuitry must be used to determine the duration between address strobe and data transfer acknowledge before issuing a bus error signal. When a bus error signal is received, the processor has two options: initiate a bus error exception sequence or try running the bus cycle again.

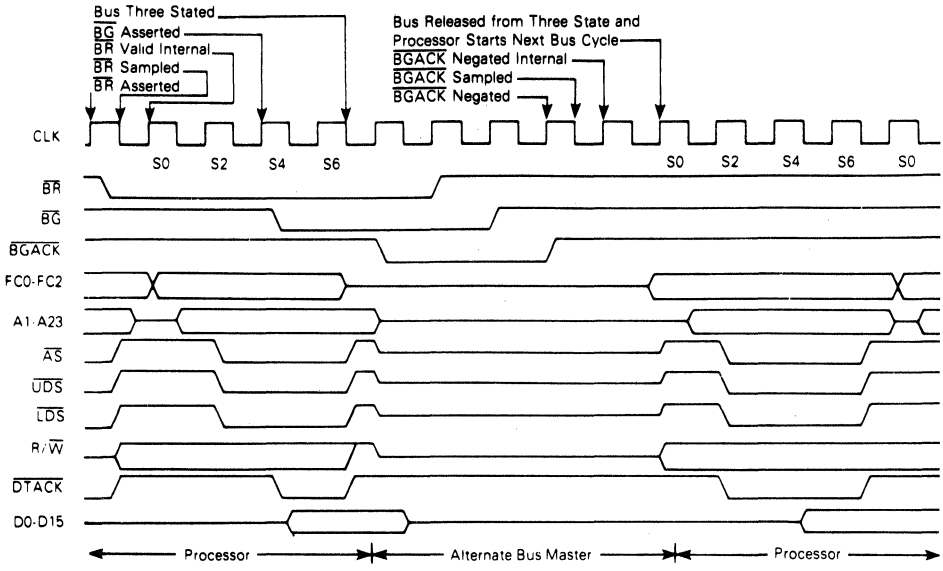


Figure 4-20. Bus Arbitration Timing Diagram—Special Case

4.2.4.1 EXCEPTION SEQUENCE. When the bus error signal is asserted, the current bus cycle is terminated. $\overline{AS}$ will be negated 2.5 clock periods after $\overline{BERR}$ is recognized. See **4.4 ASYNCHRONOUS VERSUS SYNCHRONOUS OPERATION** for more information. As long as $\overline{BERR}$ remains asserted, the data and address buses will be in the high-impedance state. When $\overline{BERR}$ is negated, the processor will begin stacking for exception processing. The sequence is composed of the following elements:

1. Stacking the program counter and status register.
2. Stacking the error information.
3. Reading the bus error vector table entry.
4. Executing the bus error handler routine.

The stacking of the program counter and the status register is the same as if an interrupt had occurred. Several additional items are stacked when a bus error occurs. These items are used to determine the nature of the error and correct it, if possible. The processor loads the new program counter from the bus error vector. A software bus error handler routine is then executed by the processor. Refer to **5.2 EXCEPTION PROCESSING** for additional information.

4.2.4.2 RE-RUNNING THE BUS CYCLE. When the processor receives a bus error signal during a bus cycle and the $\overline{HALT}$ pin is being driven by an external device, the processor enters the re-run sequence. Figure 4-21 is a timing diagram for re-running the bus cycle.

The processor terminates the bus cycle, then puts the address and data output lines in the high-impedance state. The processor remains "halted," and will not run another bus cycle until the halt signal is removed by external logic. Then the processor will re-run the previous cycle using the same function codes, the same data (for a write operation), and the same controls. The bus error signal should be removed at least one clock cycle before the halt signal is removed.

NOTE

The processor will not re-run a read-modify-write cycle. This restriction is made to guarantee that the entire cycle runs correctly and that the write operation of a test-and-set operation is performed without ever releasing $\overline{AS}$. If $\overline{BERR}$ and $\overline{HALT}$ are asserted during a read-modify-write bus cycle, a bus error operation results.

4.2.4.3 $\overline{HALT}$ OPERATION WITH NO BUS ERROR. The halt input signal to the TS68008 performs a halt/run/single-step function in a similar fashion to the 6800 halt function. The halt and run modes are somewhat self explanatory in that when the halt signal is constantly active the processor "halts" (does nothing) and when the halt signal is constantly inactive the processor "runs" (does something). $\overline{HALT}$ operation timing is shown in Figure 4-22.

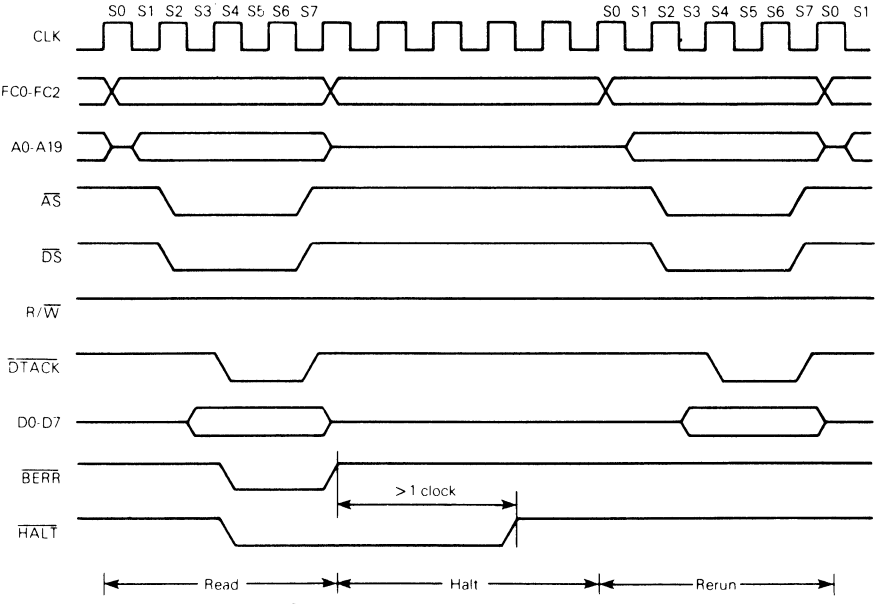


Figure 4-21. Re-Run Bus Cycle Timing Information

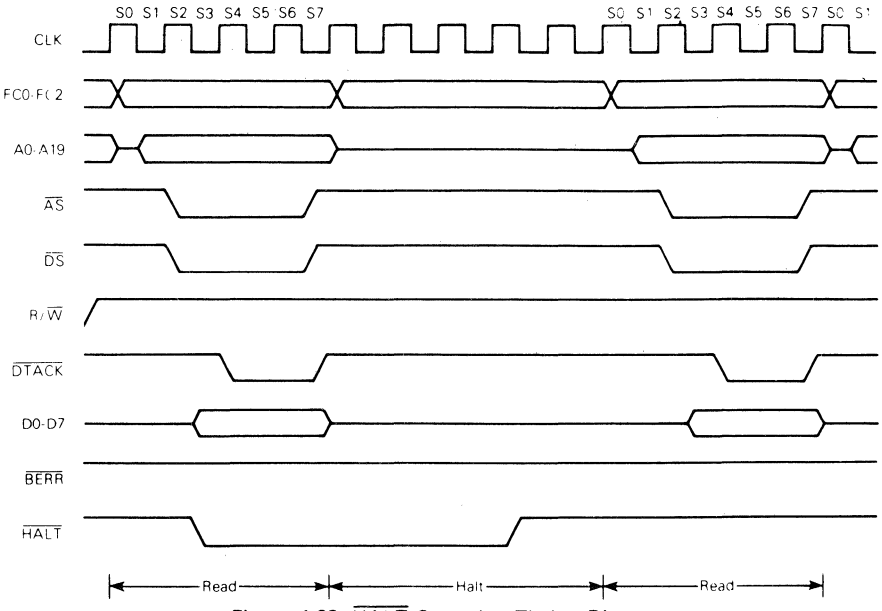


Figure 4-22. HALT Operation Timing Diagram

The single-step mode is derived from correctly timed transitions on the halt signal input. It forces the processor to execute a single bus cycle by entering the "run" mode until the processor starts a bus cycle then changing to the "halt" mode. Thus, the single-step mode allows the user to proceed through (and therefore debug) processsor operations one bus cycle at a time.

Figure 4-23 details the timing required for correct single-step operations. Some care must be exercised to avoid harmful interactions between the bus error signal and the HALT pin when using the single-cycle mode as a debugging tool. This is also true of interactions between the halt and reset lines since these can reset the machine (see **4.2.4 Reset Operation**).

When the processor completes a bus cycle after recognizing that the halt signal is active, the address and data bus signals are put in the high-impedance state.

While the processor is honoring the halt request, bus arbitration performs as usual. That is, halting has no effect on bus arbitration. It is the bus arbitration function that removes (i.e., three-states) the control signals from the bus.

The halt function and the hardware trace capability allow the hardware debugger to trace single bus cycles or single instructions at a time. These processor capabilities, along with a software debugging package, give total debugging flexibility.

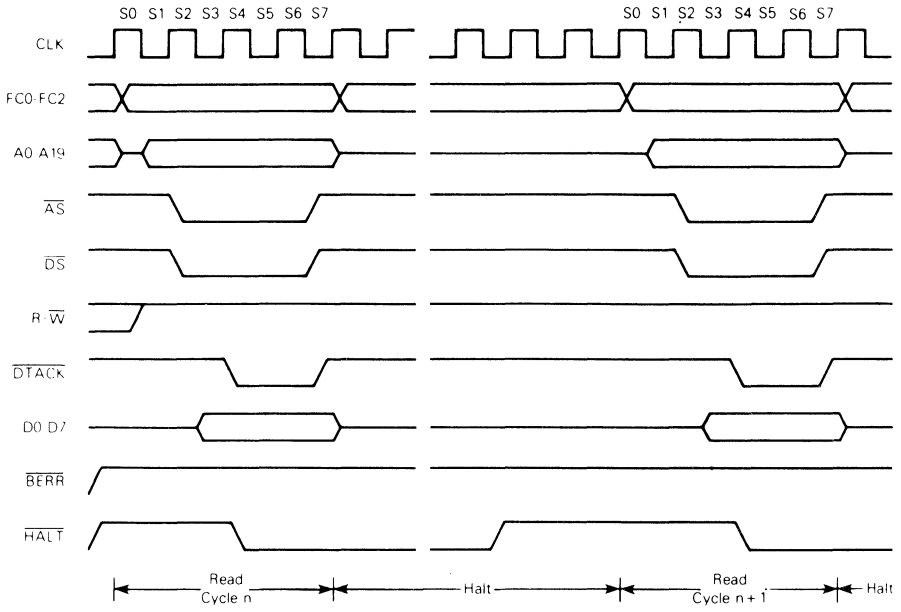


Figure 4-23. HALT Signal Single-Step Operation Timing Characteristics

4.2.4.4 DOUBLE BUS FAULTS. When a bus error exception occurs, the processor will attempt to stack several words containing information about the state of the machine. If a bus error exception occurs during the stacking operation, there have been two bus errors in a row. This is commonly referred to as a double bus fault. When a double bus fault occurs, the processor will halt. Once a bus error exception has occurred, any bus error exception occurring before the execution of the next instruction constitutes a double bus fault. Figure 4-24 is a diagram of the bus error timing.

Note that a bus cycle which is re-run does not constitute a bus error exception, and does not contribute to a double bus fault. Note also that this means that as long as the external hardware requests it, the processor will continue to re-run the same bus cycle.

The bus error pin also has an effect on processor operation after the processor receives an external reset input. The processor reads the vector table after a reset to determine the address to start program execution. If a bus error occurs while reading the vector table (or at any time before the first instruction is executed), the processor reacts as if a double bus fault has occurred and it halts. Only an external reset will start a halted processor.

4.2.5 Reset Operation

The reset signal is a bidirectional signal that allows either the processor or an external signal to reset the system. Figure 4-25 is a timing diagram for processor generated reset operation.

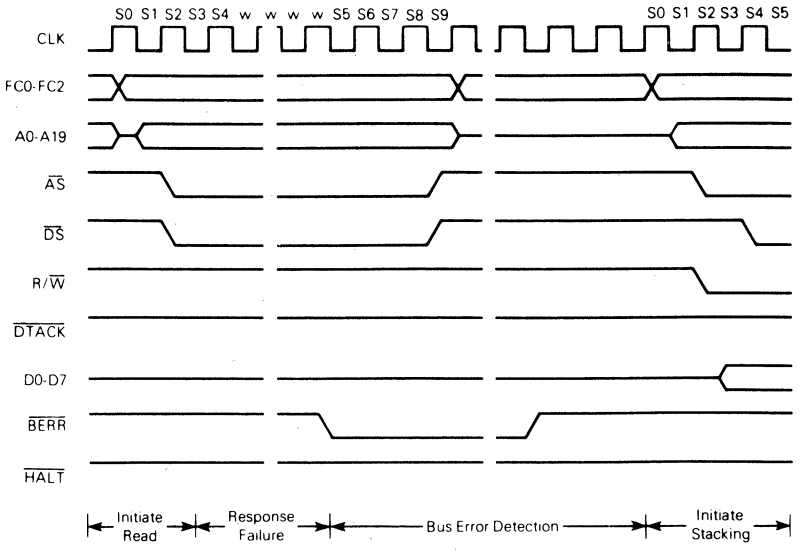


Figure 4-24. Bus Error Timing Diagram

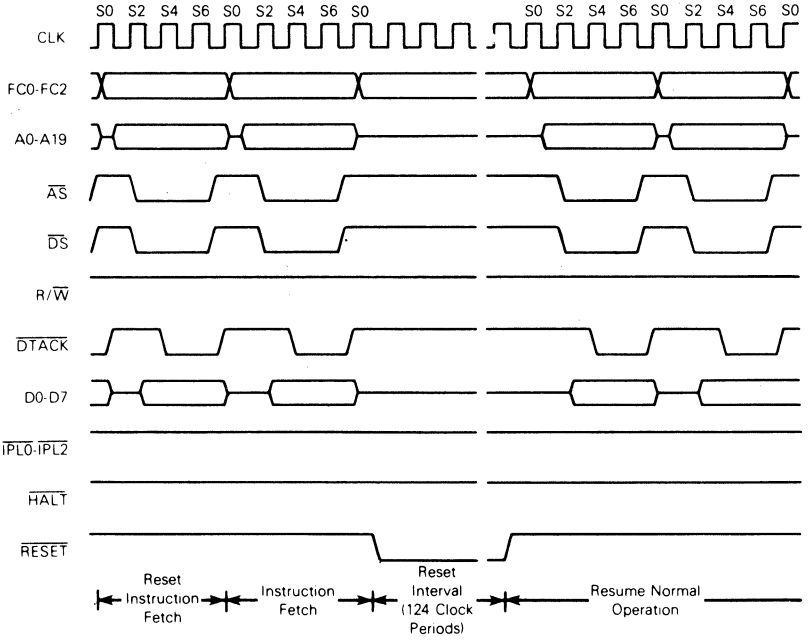


Figure 4-25. Reset Operation Timing Diagram

When the reset and halt lines are driven it is recognized as an entire system reset, including the processor. For an external reset, both the $\overline{HALT}$ and $\overline{RESET}$ lines must be asserted to ensure total reset of the processor. Timing diagram for system reset is shown in Figure 4-26. The processor responds by reading the reset vector table entry (vector number zero, address $000000) and loads it into the supervisor stack pointer (SSP). Vector table entry number one at address $000004 is read next and loaded into the program counter. The processor initializes the status register to an interrupt level of seven. No other registers are affected by the reset sequence.

When a reset instruction is executed, the processor drives the reset pin for 124 clock periods. In this case, the processor is trying to reset the rest of the system. Therefore, there is no effect on the internal state of the processor. All of the processor's internal registers and the status register are unaffected by the execution of a reset instruction. All external devices connected to the reset line will be reset at the completion of the reset instruction.

Asserting the reset and halt lines for 10 clock cycles will cause a processor reset, except when V_{CC} is initially applied to the processor. In this case, an external reset must be applied for at least 100 milliseconds allowing stabilization of the on-chip circuitry and system clock.

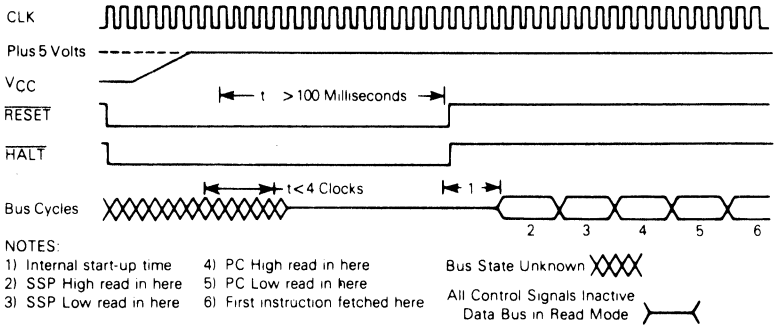


Figure 4-26. System Reset Timing Diagram

4.3 THE RELATIONSHIP OF $\overline{DTACK}$, $\overline{BERR}$, AND $\overline{HALT}$

In order to properly control termination of a bus cycle for a re-run or a bus error condition, $\overline{DTACK}$, $\overline{BERR}$, and $\overline{HALT}$ should be asserted and negated on the rising edge of the TS68008 clock. This will assure that when two signals are asserted simultaneously, the required setup time (#47) for both of them will be met during the same bus state.

This, or some equivalent precaution, should be designed external to the TS68008. Parameter #48 is intended to ensure this operation in a totally asynchronous system, and may be ignored if the above conditions are met.

The preferred bus cycle terminations may be summarized as follows (case numbers refer to Table 4-4):

Normal Termination:	$\overline{DTACK}$ occurs first (case 1).
Halt Termination:	$\overline{HALT}$ is asserted at same time, or precedes $\overline{DTACK}$ (no $\overline{BERR}$) cases 2 and 3.
Bus Error Termination:	$\overline{BERR}$ is asserted in lieu of, at same time, or preceding $\overline{DTACK}$ (Case 4); $\overline{BERR}$ negated at same time, or after $\overline{DTACK}$.
Re-Run Termination:	$\overline{HALT}$ and $\overline{BERR}$ asserted at the same time, or before $\overline{DTACK}$ (cases 5 and 6); $\overline{HALT}$ must be held at least one cycle after $\overline{BERR}$.

Table 4-4 details the resulting bus cycle termination under various combinations of control signal sequences. The negation of these same control signals under several conditions is shown in Table 4-5 ($\overline{DTACK}$ is assumed to be negated normally in all cases; for correct results, both $\overline{DTACK}$ and $\overline{BERR}$ should be negated when address strobe is negated).

EXAMPLE A:

A system uses a watch-dog timer to terminate accesses to unpopulated address space. The timer asserts $\overline{DTACK}$ and $\overline{BERR}$ simultaneously after time out (case 4).

EXAMPLE B:

A system uses error detection on RAM contents. Designer may (a) delay $\overline{DTACK}$ until data verified, and return $\overline{BERR}$ and $\overline{HALT}$ simultaneously to re-run error cycle (case 5), or if valid, return $\overline{DTACK}$; (b) delay $\overline{DTACK}$ until data verified, and return $\overline{BERR}$ at same time as $\overline{DTACK}$ if data in error (case 4); (c) return $\overline{DTACK}$ prior to data verification, as described in previous section. If data invalid, $\overline{BERR}$ is asserted (case 1) in next cycle. Error-handling software must know how to recover error cycle.

4.4 ASYNCHRONOUS VERSUS SYNCHRONOUS OPERATION

4.4.1 Asynchronous Operation

To achieve clock frequency independence at a system level, the TS68008 can be used in an asynchronous manner. This entails using only the bus handshake lines ($\overline{AS}$, $\overline{DS}$, $\overline{DTACK}$, $\overline{BERR}$, $\overline{HALT}$, and $\overline{VPA}$) to control the data transfer. Using this method, $\overline{AS}$ signals the start of a bus cycle and the data strobes are used as a condition for valid data on a write cycle. The slave device (memory or peripheral) then responds by placing the requested data on the data bus for a read cycle or latching data on a write cycle and asserting the data transfer acknowledge signal ($\overline{DTACK}$) to terminate the bus cycle. If no slave responds or the access is invalid, external control logic asserts the $\overline{BERR}$, or $\overline{BERR}$ and $\overline{HALT}$ signal to abort or rerun the bus cycle.

The $\overline{DTACK}$ signal is allowed to be asserted before the data from a slave device is valid on a read cycle. The length of time that $\overline{DTACK}$ may precede data is given as parameter #31 and it must be met in any asynchronous system to insure that valid data is latched into the processor. Notice that there is no maximum time specified from the assertion of $\overline{AS}$ to the assertion of $\overline{DTACK}$. This is because the MPU will insert wait cycles of one clock period each until $\overline{DTACK}$ is recognized.

Table 4-4. $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, and $\overline{\text{HALT}}$ Assertion Results

Case No.	Control Signal	Asserted on Rising Edge of State N	Asserted on Rising Edge of State N+2	Result
1	$\overline{\text{DTACK}}$	A	S	Normal cycle terminate and continue
	$\overline{\text{BERR}}$	NA	X	
	$\overline{\text{HALT}}$	NA	X	
2	$\overline{\text{DTACK}}$	A	S	Normal cycle terminate and halt. Continue when $\overline{\text{HALT}}$ removed.
	$\overline{\text{BERR}}$	NA	X	
	$\overline{\text{HALT}}$	A	S	
3	$\overline{\text{DTACK}}$	NA	A	Normal cycle terminate and halt. Continue when $\overline{\text{HALT}}$ removed
	$\overline{\text{BERR}}$	NA	NA	
	$\overline{\text{HALT}}$	A	S	
4	$\overline{\text{DTACK}}$	X	X	Terminate and re-run.
	$\overline{\text{BERR}}$	A	S	
	$\overline{\text{HALT}}$	NA	NA	
5	$\overline{\text{DTACK}}$	X	X	Terminate and re-run.
	$\overline{\text{BERR}}$	A	S	
	$\overline{\text{HALT}}$	A	S	
6	$\overline{\text{DTACK}}$	NA	X	Terminate and re-run when $\overline{\text{HALT}}$ removed.
	$\overline{\text{BERR}}$	NA	A	
	$\overline{\text{HALT}}$	A	S	

Legend:
N – the number of the current even bus state (e.g., S4, S6, etc.)
A – signal is asserted in this bus state
NA – signal is not asserted in this state
X – don't care
S – signal was asserted in previous state and remains asserted in this state

Table 4-5. $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ Negation Results

Conditions of Termination in Table 4-4	Control Signal	Negated on Rising Edge of State N		Negated on Rising Edge of State N+2	Results – Next Cycle
Bus Error	$\overline{\text{BERR}}$	●	or	●	Takes bus error trap
	$\overline{\text{HALT}}$	●	or	●	
Re-run	$\overline{\text{BERR}}$	●	or	●	Illegal sequence, usually traps to vector number 0
	$\overline{\text{HALT}}$	●			
Re-run	$\overline{\text{BERR}}$	●			Re-runs the bus cycle
	$\overline{\text{HALT}}$			●	
Normal	$\overline{\text{BERR}}$	●			May lengthen next cycle
	$\overline{\text{HALT}}$	●	or	●	
Normal	$\overline{\text{BERR}}$			●	If next cycle is started it will be terminated as a bus error
	$\overline{\text{HALT}}$	●	or	none	

● = Signal is negated in this bus state

4.4.2 Synchronous Operation

To allow for those systems which use the system clock as a signal to generate $\overline{DTACK}$ and other asynchronous inputs, the asynchronous input setup time is given as parameter #47. If this setup is met on an input, such as $\overline{DTACK}$, the processor is guaranteed to recognize that signal on the next falling edge of the system clock. However, the converse is not true—if the input signal does not meet the setup time it is not guaranteed not to be recognized. In addition, if $\overline{DTACK}$ is recognized on a falling edge, valid data will be latched into the processor (on a read cycle) on the next falling edge provided that the data meets the setup time given as parameter #27. Given this, parameter #31 may be ignored. Note that if $\overline{DTACK}$ is asserted, with the required setup time, before the falling edge of S4, no wait states will be incurred and the bus cycle will run at its maximum speed of four clock periods.

NOTE

During an active bus cycle, $\overline{VPA}$ and $\overline{BERR}$ are sampled on every falling edge of the clock starting with S0. $\overline{DTACK}$ is sampled on every falling edge of the clock starting with S4 and data is latched on the falling edge of S6 during a read. The bus cycle will then be terminated in S7 except when $\overline{BERR}$ is asserted in the absence of $\overline{DTACK}$, in which case it will terminate one clock cycle later in S9.

SECTION 5
PROCESSING STATES

This section describes the actions of the TS68008 which are outside the normal processing associated with the execution of instructions. The functions of the bits in the supervisor portion of the status register are covered: the supervisor/user bit, the trace enable bit, and the processor interrupt priority mask. Finally, the sequence of memory references and actions taken by the processor on exception conditions is detailed.

The TS68008 is always in one of three processing states: normal, exception, or halted. The normal processing state is that associated with instruction execution; the memory references are to fetch instructions and operands, and to store results. A special case of the normal state is the stopped state which the processor enters when a STOP instruction is executed. In this state, no further memory references are made.

The exception processing state is associated with interrupts, trap instructions, tracing, and other exceptional conditions. The exception may be internally generated by an instruction or by an unusual condition arising during the execution of an instruction. Externally, exception processing can be forced by an interrupt, by a bus error, or by a reset. Exception processing is designed to provide an efficient context switch so that the processor may handle unusual conditions.

The halted processing state is an indication of catastrophic hardware failure. For example, if during the exception processing of a bus error another bus error occurs, the processor assumes that the system is unusable and halts. Only an external reset can restart a halted processor. Note that a processor in the stopped state is not in the halted state, nor vice versa.

5.1 PRIVILEGE STATES

The processor operates in one of two states of privilege: the "user" state or the "supervisor" state. The privilege state determines which operations are legal, is used by the external memory management device to control and translate accesses, and is used to choose between the supervisor stack pointer and the user stack pointer in instruction references.

The privilege state is a mechanism for providing security in a computer system. Programs should access only their own code and data areas, and ought to be restricted from accessing information which they do not need and must not modify.

The privilege mechanism provides security by allowing most programs to execute in user state. In this state, the accesses are controlled, and the effects on other parts of the system are limited. The operating system executes in the supervisor state, has access to all resources, and performs the overhead tasks for the user state programs.

5.1.1 Supervisor State

The supervisor state is the higher state of privilege. For instruction execution, the supervisor state is determined by the S bit of the status register; if the S bit is asserted (high) or exception processing is invoked, the processor is in the supervisor state. All instructions can be executed in the supervisor state. The bus cycles generated by instructions executed in the supervisor state are classified as supervisor references. While the processor is in the supervisor privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly access the supervisor stack pointer.

5.1.2 User State

The user state is the lower state of privilege and is controlled by the S bit of the status register. If the S bit is negated (low), the processor is executing instructions in the user state. The bus cycles generated by an instruction executed in the user state are classified as user state references. This allows an external memory management device to translate the address and to control access to protected portions of the address space. While the processor is in the user privilege state, those instructions which use either the system stack pointer implicitly, or address register seven explicitly, access the user stack pointer.

Most instructions execute the same in user state as in the supervisor state. However, some instructions which have important system effects are made privileged. User programs are not permitted to execute the STOP instruction, or the RESET instruction. To ensure that a user program cannot enter the supervisor state except in a controlled manner, the instructions which modify the whole status register are privileged. To aid in debugging programs which are to be used as operating systems, the move to user stack pointer (MOVE USP) and move from user stack pointer (MOVE from USP) instructions are also privileged.

5.1.3 Privilege State Changes

Once the processor is in the user state and executing instructions, only exception processing can change the privilege state. During exception processing, the current setting of the S bit of the status register is saved and the S bit is asserted, putting the processor in the supervisor state. Therefore, when instruction execution resumes at the address specified to process the exception, the processor is in the supervisor privilege state.

5.1.4 Reference Classification

When the processor makes a reference, it classifies the kind of reference being made, using the encoding on the three function code output lines. This allows external translation of addresses, control of access, and differentiation of special processor states, such as interrupt acknowledge. Table 5-1 lists the classification of references.

5.2 EXCEPTION PROCESSING

Before discussing the details of interrupts, traps, and tracing, a general description of exception processing is in order. The processing of an exception occurs in four steps, with variations for different exception causes. During the first step, a temporary copy of the status register is made, and the status register is set for exception processing. In the second step the exception vector is determined, and the third step is the saving of the current processor context. In the fourth step a new context is obtained, and the processor switches to instruction processing.

Table 5-1. Reference Classification

Function Code Output			Reference Class
FC2	FC1	FC0	
0	0	0	(Unassigned)
0	0	1	User Data
0	1	0	User Program
0	1	1	(Unassigned)
1	0	0	(Unassigned)
1	0	1	Supervisor Data
1	1	0	Supervisor Program
1	1	1	Interrupt Acknowledge

5.2.1 Exception Vectors

Exception vectors are memory locations from which the processor fetches the address of a routine which will handle that exception. All exception vectors are two words in length (Figure 5-1), except for the reset vector, which is four words. All exception vectors lie in the supervisor data space, except for the reset vector which is in the supervisor program space. A vector number is an 8-bit number which, when multiplied by four, gives the address of an exception vector. Vector numbers are generated internally or externally, depending on the cause of the exception. In the case of vectored interrupts, during the interrupt acknowledge bus cycle, a peripheral provides an 8-bit vector number (Figure 5-2) to the processor on data bus lines D0 through D7. The processor translates the vector number into a full 32-bit address, as shown in Figure 5-3. The memory layout for exception vectors is given in Table 5-2.

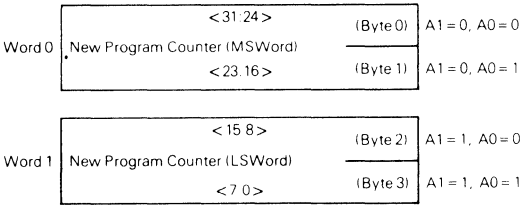


Figure 5-1. Format of Vector Table Entries

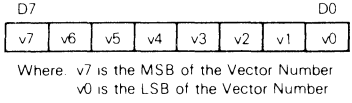


Figure 5-2. Vector Number Format

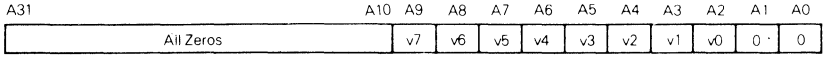


Figure 5-3. Vector Number Translated to an Address

Table 5-2. Vector Table

Vector Number(s)	Address			Assignment
	Dec	Hex	Space	
0	0	000	SP	Reset: Initial SSP
–	4	004	SP	Reset: Initial PC
2	8	008	SD	Bus Error
3	12	00C	SD	Address Error
4	16	010	SD	Illegal Instruction
5	20	014	SD	Zero Divide
6	24	018	SD	CHK Instruction
7	28	01C	SD	TRAPV Instruction
8	32	020	SD	Privilege Violation
9	36	024	SD	Trace
10	40	028	SD	Line 1010 Emulator
11	44	02C	SD	Line 1111 Emulator
12*	48	030	SD	(Unassigned, Reserved)
13*	52	034	SD	(Unassigned, Reserved)
14*	56	038	SD	(Unassigned, Reserved)
15	60	03C	SD	Uninitialized Interrupt Vector
16-23*	64	04C	SD	(Unassigned, Reserved)
	95	05F		–
24	96	060	SD	Spurious Interrupt
25	100	064	SD	Level 1 Interrupt Autovector
26	104	068	SD	Level 2 Interrupt Autovector
27	108	06C	SD	Level 3 Interrupt Autovector
28	112	070	SD	Level 4 Interrupt Autovector
29	116	074	SD	Level 5 Interrupt Autovector
30	120	078	SD	Level 6 Interrupt Autovector
31	124	07C	SD	Level 7 Interrupt Autovector
32-47	128	080	SD	TRAP Instruction Vectors
	191	0BF		–
48-63*	192	0C0	SD	(Unassigned, Reserved)
	255	0FF		–
64-255	256	100	SD	User Interrupt Vectors
	1023	3FF		–

*Vector numbers 12, 13, 14, 16 through 23, and 48 through 63 are reserved for future enhancements by THOMSON SEMICONDUCTEURS. No user peripheral devices should be assigned these numbers.

As shown in Table 5-2, the memory layout is 512 words long (1024 bytes). It starts at address 0 and proceeds through address 1023. This provides 255 unique vectors; some of these are reserved for TRAPS and other system functions. Of the 255, there are 192 reserved for user interrupt vectors. However, there is no protection on the first 64 entries, so user interrupt vectors may overlap at the discretion of the systems designer.

5.2.2 Kinds of Exceptions

Exceptions can be generated by either internal or external causes. The externally generated exceptions are the interrupts and the bus error and reset requests. The interrupts are requests from peripheral devices for processor action while the bus error and reset inputs are used for access control and processor restart. The internally generated exceptions come from instructions, or from address errors or tracing. The trap (TRAP), trap on overflow (TRAPV), check register against

bounds (CHK), and divide (DIV) instructions all can generate exceptions as part of their instruction execution. In addition, illegal instructions, word fetches from odd addresses and privilege violations cause exceptions. Tracing behaves like a very high priority, internally generated interrupt after each instruction execution.

5.2.3 Exception Processing Sequence

Exception processing occurs in four identifiable steps. In the first step, an internal copy is made of the status register. After the copy is made, the S bit is asserted, putting the processor into the supervisor privilege state. Also, the T bit is negated which will allow the exception handler to execute unhindered by tracing. For the reset and interrupt exceptions, the interrupt priority mask is also updated.

In the second step, the vector number of the exception is determined. For interrupts, the vector number is obtained by a processor fetch, classified as an interrupt acknowledge. For all other exceptions, internal logic provides the vector number. This vector number is then used to generate the address of the exception vector.

The third step is to save the current processor status, except for the reset exception. The current program counter value and the saved copy of the status register are stacked using the supervisor stack pointer. The program counter value stacked usually points to the next unexecuted instruction, however, for bus error and address error, the value stacked for the program counter is unpredictable, and may be incremented from the address of the instruction which caused the error. Additional information defining the current context is stacked for the bus error and address error exceptions.

The last step is the same for all exceptions. The new program counter value is fetched from the exception vector. The processor then resumes instruction execution. The instruction at the address given in the exception vector is fetched, and normal instruction decoding and execution is started.

5.2.4 Multiple Exceptions

These paragraphs describe the processing which occurs when multiple exceptions arise simultaneously. Exceptions can be grouped according to their occurrence and priority. The group 0 exceptions are reset, address error, and bus error. These exceptions cause the instruction currently being executed to be aborted, and the exception processing to commence within two clock cycles.

The group 1 exceptions are trace and interrupt, as well as the privilege violations and illegal instructions. The trace and interrupt exceptions allow the current instruction to execute to completion, but pre-empt the execution of the next instruction by forcing exception processing to occur (privilege violations and illegal instructions are detected when they are the next instruction to be executed). The group 2 exceptions occur as part of the normal processing of instructions. The TRAP, TRAPV, CHK, and zero divide exceptions are in this group. For these exceptions, the normal execution of an instruction may lead to exception processing.

Group 0 exceptions have highest priority, while group 2 exceptions have lowest priority. Within group 0, reset has highest priority, followed by address error and then bus error. Within group 1, trace has priority over external interrupts, which in turn takes priority over illegal instruction and privilege violation. Since only one instruction can be executed at a time, there is no priority relation within group 2.

The priority relation between two exceptions determines which is taken, or taken first, if the conditions for both arise simultaneously. Therefore, if a bus error occurs during a TRAP instruction, the bus error takes precedence, and the TRAP instruction processing is aborted. In another example, if an interrupt request occurs during the execution of an instruction while the T bit is asserted, the trace exception has priority, and is processed first. Before instruction processing resumes, however, the interrupt exception is also processed, and instruction processing commences finally in the interrupt handler routine. A summary of exception grouping and priority is given in Table 5-3.,

Table 5-3. Exception Grouping and Priority

Group	Exception	Processing
0	Reset Address Error Bus Error	Exception processing begins within two clock cycles
1	Trace Interrupt Illegal Privilege	Exception processing begins before the next instruction
2	TRAP, TRAPV, CHK, Zero Divide	Exception processing is started by normal instruction execution

5.3 EXCEPTION PROCESSING DETAILED DISCUSSION

Exceptions have a number of sources, and each exception has processing which is peculiar to it. The following paragraphs detail the sources of exceptions, how each arises, and how each is processed.

5.3.1 Reset

The reset input provides the highest exception level. The processing of the reset signal is designed for system initiation, and recovery from catastrophic failure. Any processing in progress at the time of the reset is aborted and cannot be recovered. The processor is forced into the supervisor state, and the trace state is forced off. The processor interrupt priority mask is set at level seven. The vector number is internally generated to reference the reset exception vector at location 0 in the supervisor program space. Because no assumptions can be made about the validity of register contents, in particular the supervisor stack pointer, neither the program counter nor the status register is saved. The address contained in the first two words of the reset exception vector is fetched as the initial supervisor stack pointer, and the address in the last two words of the reset exception vector is fetched as the initial program counter. Finally, instruction execution is started at the address in the program counter. The power-up/restart code should be pointed to by the initial program counter.

The reset instruction does not cause loading of the reset vector, but does assert the reset line to reset external devices. This allows the software to reset the system to a known state and then continue processing with the next instruction.

5.3.2 Interrupts

Seven levels of interrupts are provided by the 68000 architecture. The TS68008 supports three interrupt levels: two, five, and seven, level seven being the highest. Devices may be chained externally within interrupt priority levels, allowing an unlimited number of peripheral devices to interrupt

the processor. The status register contains a 3-bit mask which indicates the current processor priority, and interrupts are inhibited for all priority levels less than or equal to the current processor priority.

An interrupt request is made to the processor by encoding the interrupt request level on the interrupt request lines; a zero indicates no interrupt request. Interrupt requests arriving at the processor do not force immediate exception processing, but are made pending. Pending interrupts are detected between instruction executions. If the priority of the pending interrupt is lower than or equal to the current processor priority, execution continues with the next instruction and the interrupt exception processing is postponed. (The recognition of level seven is slightly different, as explained in the following paragraph.)

If the priority of the pending interrupt is greater than the current processor priority, the exception processing sequence is started. First a copy of the status register is saved, and the privilege state is set to supervisor, tracing is suppressed, and the processor priority level is set to the level of the interrupt being acknowledged. The processor fetches the vector number from the interrupting device, classifying the reference as an interrupt acknowledge and displaying the level number of the interrupt being acknowledged on the address bus. If external logic requests an automatic vectoring, the processor internally generates a vector number which is determined by the interrupt level number. If external logic indicates a bus error, the interrupt is taken to be spurious, and the generated vector number references the spurious interrupt vector. The processor then proceeds with the usual exception processing, saving the program counter and status register on the supervisor stack. The saved value of the program counter is the address of the instruction which would have been executed had the interrupt not been present. The content of the interrupt vector whose vector number was previously obtained is fetched and loaded into the program counter, and normal instruction execution commences in the interrupt handling routine. A flowchart for the interrupt acknowledge sequence is given in Figure 5-4, a timing diagram is given in Figure 5-5, and the interrupt processing sequence is shown in Figure 5-6.

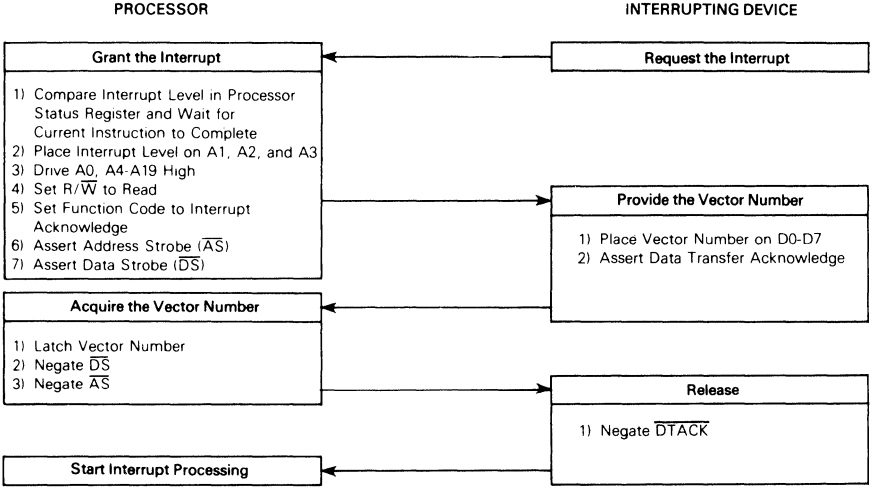


Figure 5-4. Vector Acquisition Flowchart

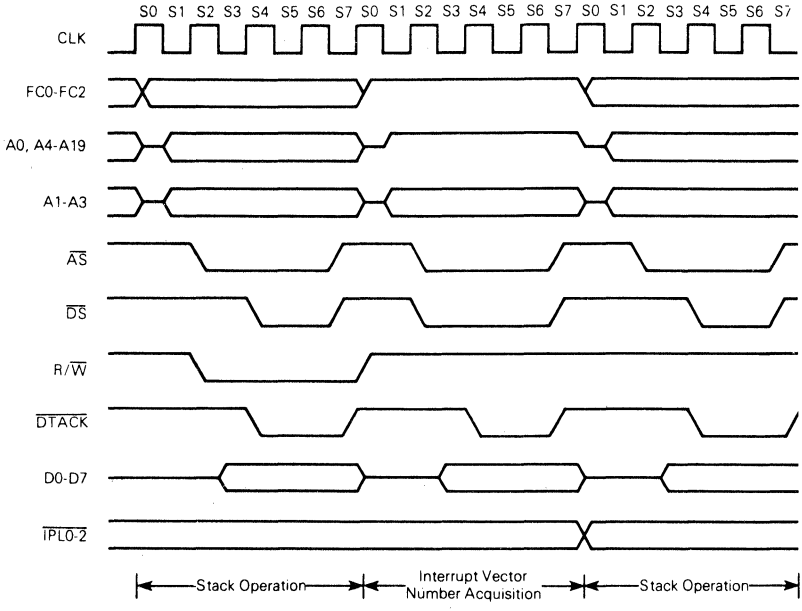


Figure 5-5. Interrupt Acknowledge Cycle

1. Acquire vector number via interrupt acknowledge.
2. Convert vector number to a full 32-bit address.
3. Stack the SR and PC by successive write cycles. Refer to Figure 4-7 for word write cycle operation.
4. Place vector table address on A0-A19. Refer to Figure 5-3 for address translation.
5. Read upper half of program counter (PC). Refer to Figure 4-3 for word read cycle operation.
6. Increment vector table address by 2 and place it on A0-A19.
7. Read lower half of program counter (PC).
8. Load new program counter (PC).
9. Place contents of PC on A0-A19.
10. Read first instruction of service routine.

Figure 5-6. Interrupt Processing Sequence

Priority level seven is a special case. Level seven interrupts cannot be inhibited by the interrupt priority mask, thus providing a "non-maskable interrupt" capability. An interrupt is generated each time the interrupt request level changes from some lower level to level seven. Note that a level seven interrupt may still be caused by the level comparison if the request level is a seven and the processor priority is set to a lower level by an instruction.

5.3.3 Uninitialized Interrupt

An interrupting device asserts $\overline{VPA}$ or provides an interrupt vector during an interrupt acknowledge cycle to the TS68008. If the vector register of the peripheral has not been initialized, the responding 68000 Family peripheral will provide vector 15 ($0F), the uninitialized interrupt vector. This provides a uniform way to recover from a programming error.

5.3.4 Spurious Interrupt

If during the interrupt acknowledge cycle no device responds by asserting $\overline{DTACK}$ or $\overline{VPA}$, the bus error line should be asserted to terminate the vector acquisition. The processor separates the processing of this error from bus error by fetching the spurious interrupt vector instead of the bus error vector. The processor then proceeds with the usual exception processing.

5.3.5 Instruction Traps

Traps are exceptions caused by instructions. They arise either from processor recognition of abnormal conditions during instruction execution. or from use of instructions whose normal behavior is trapping. The TRAP instruction always forces an exception, and is useful for implementing system calls for user programs. The TRAPV and CHK instructions force an exception if the user program detects a runtime error, which may be an arithmetic overflow or a subscript out of bounds. The signed divide (DIVS) and unsigned divide (DIVU) instructions will force an exception if a division operation is attempted with a divisor of zero.

5.3.6 Illegal and Unimplemented Instructions

"Illegal instruction" is the term used to refer to any of the word bit patterns which are not the bit pattern of the first word of a legal instruction. During instruction execution, if such an instruction is fetched, an illegal instruction exception occurs. THOMSON SEMICONDUCTEURS reserves the right to define instructions whose opcodes may be any of the illegal instructions. Three bit patterns will always force an illegal instruction trap on all 68000 Family compatible microprocessors. They are : $4AFA, $4AFB and $4AFC. Two of the patterns, $4AFA and $4AFB, are reserved for THOMSON SEMICONDUCTEURS system products. The third pattern, $4AFC, is reserved for customer use.

Word patterns with bits 15 through 12 equaling 1010 or 1111 are distinguished as unimplemented instructions and separate exception vectors are given to these patterns to permit efficient emulation. This facility allows the operating system to detect program errors, or to emulate unimplemented instructions in software.

5.3.7 Privilege Violations

In order to provide system security, various instructions are privileged. An attempt to execute one of the privileged instructions while in the user state will cause an exception. The privileged instructions are:

STOP	AND Immediate to SR
RESET	EOR Immediate to SR
RTE	OR Immediate to SR
MOVE to SR	MOVE USP

5.3.8 Tracing

To aid in program development, the TS68008 includes a facility to allow instruction-by-instruction tracing. In the trace state, after each instruction is executed an exception is forced, allowing a debugging program to monitor the execution of the program under test.

The trace facility uses the T bit in the supervisor portion of the status register. If the T bit is negated (off), tracing is disabled, and instruction execution proceeds from instruction to instruction as normal. If the T bit is asserted (on) at the beginning of the execution of an instruction, a trace exception will be generated after the execution of that instruction is completed. If the instruction is not executed, either because an interrupt is taken, or the instruction is illegal or privileged, the trace exception does not occur. The trace exception also does not occur if the instruction is aborted by a reset, bus error, or address error exception. If the instruction is indeed executed and an interrupt is pending on completion, the trace exception is processed before the interrupt exception. If, during the execution of the instruction, an exception is forced by that instruction, the forced exception is processed before the trace exception.

As an extreme illustration of the above rules, consider the arrival of an interrupt during the execution of a TRAP instruction while tracing is enabled. First the trap exception is processed, then the trace exception, and finally the interrupt exception. Instruction execution resumes in the interrupt handler routine.

5.3.9 Bus Error

Bus error exceptions occur when the external logic requests that a bus error be processed by an exception. The current bus cycle which the processor is making is then aborted. Regardless of whether the processor was doing instruction or exception processing, that processing is terminated, and the processor immediately begins exception processing.

Exception processing for bus error follows the usual sequence of steps. The status register is copied, the supervisor state is entered, and the trace state is turned off. The vector number is generated to refer to the bus error vector. Since the processor was not between instructions when the bus error exception request was made, the context of the processor is more detailed. To save more of this context, additional information is saved on the supervisor stack (refer to Figure 5-7). The program counter and the copy of the status register are of course saved. The value saved for the program counter is advanced by some amount, two to ten bytes beyond the address of the first word of the instruction which made the reference causing the bus error. If the bus error occurred during the fetch of the next instruction, the saved program counter has a value in the vicinity of the current instruction, even if the current instruction is a branch, a jump, or a return instruction. Besides the usual information, the processor saves its internal copy of the first word of the instruction being processed, and the address which was being accessed by the aborted bus cycle. Specific information about the access is also saved: whether it was a read or a write, whether the processor was processing an instruction or not, and the classification displayed on the function code outputs when the bus error occurred. The processor is processing an instruction if it is in the normal state or processing a group 2 exception; the processor is not processing an instruction if it is processing a group 0 or a group 1 exception. Figure 5-7 illustrates how this information is organized on the supervisor stack. Although this information is not sufficient in general to effect full recovery from the bus error, it does allow software diagnosis. Finally, the processor commences instruction processing at the address contained in the vector. It is the responsibility of the error handler routine to clean up the stack and determine where to continue execution.

If a bus error occurs during the exception processing for a bus error, address error, or reset, the processor is halted, and all processing ceases. This simplifies the detection of catastrophic system failure, since the processor removes itself from the system rather than destroy all memory contents. Only the $\overline{\text{RESET}}$ pin can restart a halted processor.

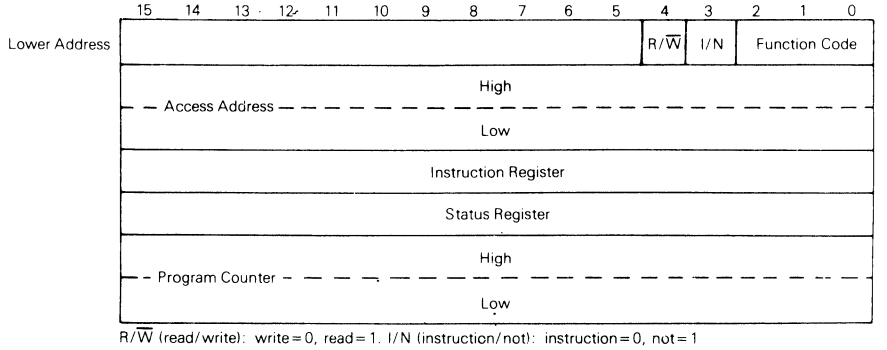


Figure 5-7. Supervisor Stack Order (Group 0)

5.3.10 Address Error

Address error exceptions occur when the processor attempts to access a word or a long word operand or an instruction at an odd address. When the TS68008 detects an address error it prevents assertion of $\overline{DS}$ but asserts $\overline{AS}$ to provide proper bus arbitration support. The effect is much like an internally generated bus error, in that the bus cycle is aborted, and the processor ceases whatever processing it is currently doing and begins exception processing. After exception processing commences, the sequence is the same as that for bus error including the information that is stacked, except that the vector number refers to the address error vector instead. Likewise, if an address error occurs during the exception processing for a bus error, address error, or reset, the processor is halted. As shown in Figure 5-8, an address error will execute a short bus cycle followed by exception processing.

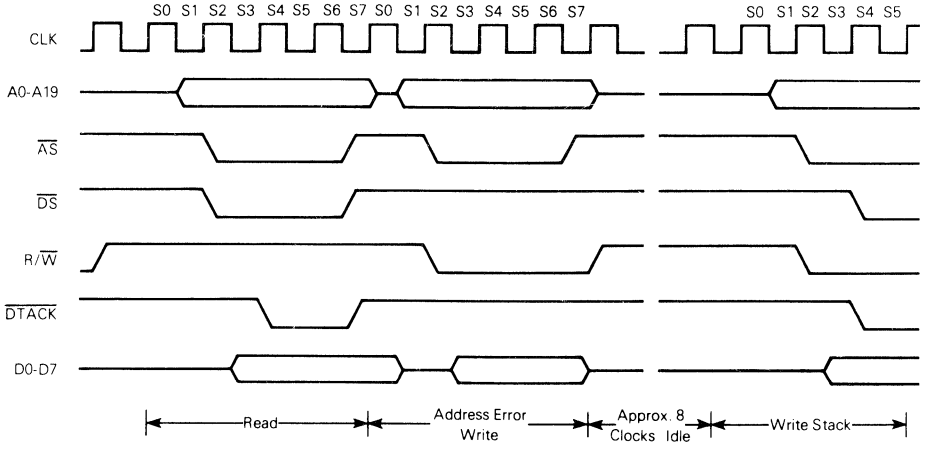


Figure 5-8. Address Error Timing

SECTION 6
INTERFACE WITH 6800 PERIPHERALS

THOMSON SEMICONDUCTEURS' extensive line of 6800 peripherals are compatible with the TS68008. Some of these devices that are particularly useful are :

EF6821 Peripheral Interface Adapter
EF6840 Programmable Timer Module
EF9345,EF9367 CRT Controllers
EF6850 Asynchronous Communications Interface Adapter
EF6852 Synchronous Serial Data Adapter
EF6854 Advanced Data Link Controller

To interface the synchronous 6800 peripherals with the asynchronous TS68008, the processor modifies its bus cycle to meet the 6800 cycle requirements whenever an 6800 device address is detected. This is possible since both processors use memory mapped I/O. Figure 6-1 is a flowchart of the interface operation between the processor and 6800 devices.

6.1 DATA TRANSFER OPERATION

Two signals on the processor provide the 6800 interface. They are : enable (E), and valid peripheral address (VPA). In addition, a valid memory address (VMA) signal must be provided (see **4.1.7 6800 Peripheral Control**). Enable corresponds to the E signal in existing 6800 systems. The E clock frequency is one tenth of the incoming TS68008 clock frequency. The timing of E allows 1 megahertz peripherals to be used with an 8 megahertz TS68008. Enable has a 60/40 duty cycle ; that is, it is low for six input clocks and high for four input clocks.

6800 cycle timing is given in Section 8. At state zero in the cycle, the address bus is in the high-impedance state. A function code is asserted on the function code output lines. One-half clock later, in state one, the address bus is released from the high-impedance state.

During state two, the address strobe ($\overline{AS}$) is asserted to indicate that there is a valid address on the address bus. If the bus cycle is a read cycle, the data strobe is also asserted in state two. If the bus cycle is a write cycle the read/write (R/$\overline{W}$) signal is switched to low (write) during state two. One half clock later, in state three, the write data is placed on the data bus, and in state four the data strobe is issued to indicate valid data on the data bus. The processor now inserts wait states until it recognizes the assertion of $\overline{VPA}$.

The $\overline{VPA}$ input signals the processor that the address on the bus is the address of an 6800 device (or an area reserved for 6800 devices) and that the bus should conform to the transfer characteristics of the 6800 bus. Valid peripheral address is derived by decoding the address bus, conditioned by address strobe. Chip select for the 6800 peripherals should be derived by decoding the address bus conditioned by $\overline{VMA}$ (not $\overline{AS}$).

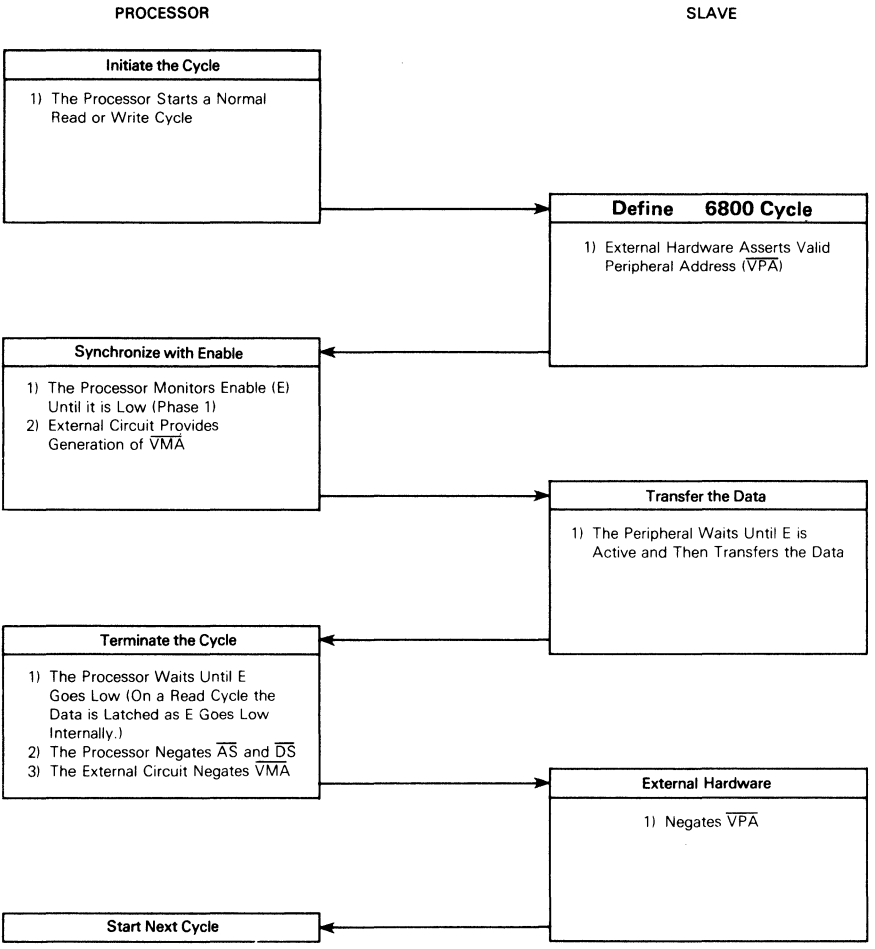


Figure 6-1. 6800 Cycle Flowchart

After recognition of $\overline{VPA}$, the processor assures that the enable (E) is low, by waiting if necessary. Valid memory address (provided by an external circuit similar to that of Figure 4-2) is then used as part of the chip select equation of the peripheral. This ensures that the 6800 peripherals are selected and deselected at the correct time. The peripheral now runs its cycle during the high portion of the E signal. Figure 6-2 depicts the 6800 cycle timing using the VMA generation circuit shown in Figure 4-2. This cycle length is dependent strictly upon when $\overline{VPA}$ is asserted in relationship to the E clock.

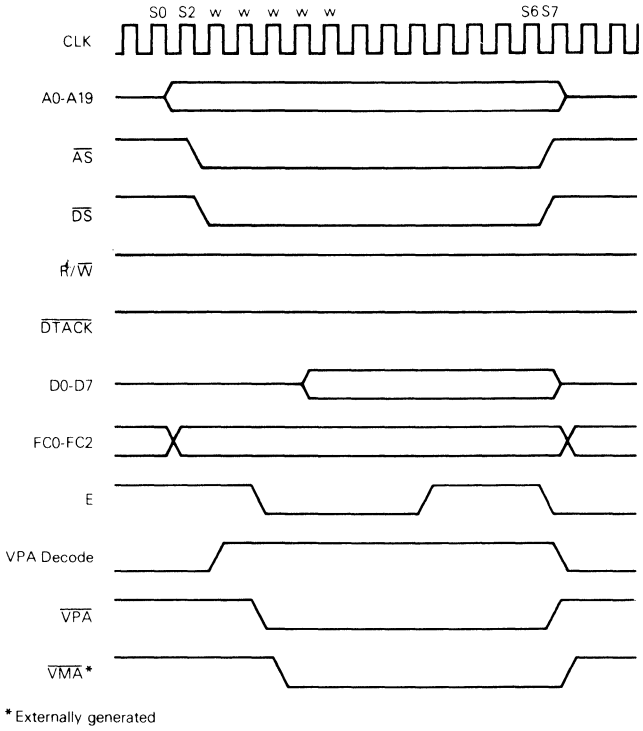


Figure 6-2. 6800 Cycle Timing

During a read cycle, the processor latches the peripheral data in state six. For all cycles, the processor negates the address and data strobes one half clock cycle later in state seven, and the enable signal goes low at this time. Another half clock later, the address bus is put in the high-impedance state. During a write cycle, the data bus is put in the high-impedance state and the read/write signal is switched high. The peripheral logic must remove $\overline{VPA}$ within one clock after address strobe is negated.

$\overline{DTACK}$ should not be asserted while $\overline{VPA}$ is asserted. Notice that $\overline{VMA}$ is active low, contrasted with the active high 6800 VMA. Refer to Figure 4-2.

6.2 AC ELECTRICAL SPECIFICATIONS

The electrical specifications for interfacing the TS68008 to 6800 Family peripherals are located in Section 8.

6.3 INTERRUPT INTERFACE OPERATION

During an interrupt acknowledge cycle while the processor is fetching the vector, the $\overline{VPA}$ is asserted, the TS68008 will complete a normal 6800 read cycle as shown in Figure 6-3. The processor will then use an internally generated vector that is a function of the interrupt being serviced. This process is known as autovectoring. The seven autovectors are vector numbers 25 through 31 (decimal).

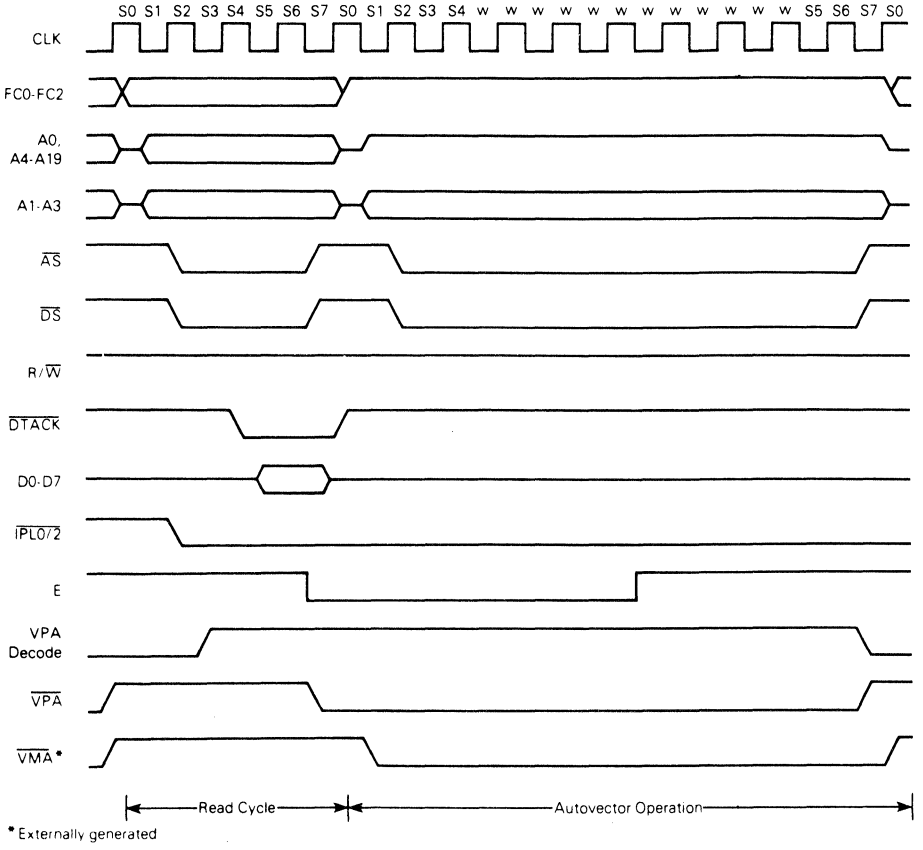


Figure 6-3. Autovector Operation Timing Diagram

Autovectoring operates in the same fashion (but is not restricted to) the 6800 interrupt sequence. The basic difference is that there are six normal interrupt vectors and one NMI type vector. As with both the 6800 and the TS68008's normal vectored interrupt, the interrupt service routine can be located anywhere in the address space. This is due to the fact that while the vector numbers are fixed, the contents of the vector table entries are assigned by the user.

Since $\overline{VMA}$ is asserted during autovectoring, the 6800 peripheral address decoding should prevent unintended accesses.

SECTION 7
INSTRUCTION SET AND EXECUTION TIMES

7.1 INSTRUCTION SET

The following paragraphs provide information about the addressing categories and instruction set of the TS68008.

7.1.1 Addressing Categories

Effective address modes may be categorized by the ways in which they may be used. The following classifications will be used in the instruction definitions.

Data — If an effective address mode by be used to refer to data operands, it is considered a data addressing effective address mode.

Memory — If an effective address mode may be used to refer to memory operands, it is considered a memory addressing effective address mode.

Alterable — If an effective address mode may be used to refer to alterable (writeable) operands, it is considered an alterable addressing effective address mode.

Control — If an effective address mode may be used to refer to memory operands without an associated size, it is considered a control addressing effective address mode.

These categories may be combined, so that additional, more restrictive, classifications may be defined. For example, the instruction descriptions use such classifications as alterable memory or data alterable. The former refers to those addressing modes which are both alterable and memory addresses, and the latter refers to addressing modes which are both data and alterable. Table 7-1 shows the various categories to which each of the effective address modes belong. Table 7-2 is the instruction set summary.

Table 7-1. Effective Addressing Mode Categories

Effective Address Modes	Mode	Register	Data	Addressing Categories		
				Memory	Control	Alterable
Dn	000	Register Number	X	–	–	X
An	001	Register Number	–	–	–	X
(An)	010	Register Number	X	X	X	X
(An)+	011	Register Number	X	X	–	X
–(An)	100	Register Number	X	X	–	X
d(An)	101	Register Number	X	X	X	X
d(An, ix)	110	Register Number	X	X	X	X
xxx.W	111	000	X	X	X	X
xxx.L	111	001	X	X	X	X
d(PC)	111	010	X	X	X	–
d(PC, ix)	111	011	X	X	X	–
#xxx	111	100	X	X	–	–

Table 7-2. Instruction Set (Sheet 1 of 2)

Mnemonic	Description	Operation	Condition Codes				
			X	N	Z	V	C
ABCD	Add Decimal with Extend	(Destination)$_{10}$ + (Source)$_{10}$ + x → Destination	*	U	*	U	*
ADD	Add Binary	(Destination) + (Source) → Destination	*	*	*	*	*
ADDA	Add Address	(Destination) + (Source) → Destination	–	–	–	–	–
ADDI	Add Immediate	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDQ	Add Quick	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDX	Add Extended	(Destination) + (Source) + x → Destination	*	*	*	*	*
AND	AND Logical	(Destination) ∧ (Source) → Destination	–	*	*	0	0
ANDI	AND Immediate	(Destination) ∧ Immediate Data → Destination	–	*	*	0	0
ASL, ASR	Arithmetic Shift	(Destination) Shifted by <count> → Destination	*	*	*	*	*
B$_{CC}$	Branch Conditionally	If $_{CC}$ then PC + d → PC	–	–	–	–	–
BCHG	Test a Bit and Change	~(<bit number>) OF Destination → Z ~(<bit number>) OF Destination → <bit number> OF Destination	–	–	*	–	–
BCLR	Test a Bit and Clear	~(<bit number>) OF Destination → Z 0 → <bit number> → OF Destination	–	–	*	–	–
BRA	Branch Always	PC + displacement → PC	–	–	–	–	–
BSET	Test a Bit and Set	~(<bit number>) OF Destination → Z 1 → <bit number> OF Destination	–	–	*	–	–
BSR	Branch to Subroutine	PC → – (SP), PC + d → PC	–	–	–	–	–
BTST	Test a Bit	~(<bit number>) OF Destination → Z	–	–	*	–	–
CHK	Check Register against Bounds	If Dn <0 or Dn> (<ea>) then TRAP	–	*	U	U	U
CLR	Clear an Operand	0 → Destination	–	0	1	0	0
CMP	Compare	(Destination) – (Source)	–	*	*	*	*
CMPA	Compare Address	(Destination) – (Source)	–	*	*	*	*
CMPI	Compare Immediate	(Destination) – Immediate Data	–	*	*	*	*
CMPM	Compare Memory	(Destination) – (Source)	–	*	*	*	*
DB$_{CC}$	Test Condition, Decrement and Branch	If ~$_{CC}$ then Dn – 1 → Dn; if Dn ≠ – 1 then PC + d → PC	–	–	–	–	–
DIVS	Signed Divide	(Destination)/(Source) → Destination	–	*	*	*	0
DIVU	Unsigned Divide	(Destination)/(Source) → Destination	–	*	*	*	0
EOR	Exclusive OR Logical	(Destination) ⊕ (Source) → Destination	–	*	*	0	0
EORI	Exclusive OR Immediate	(Destination) ⊕ Immediate Data → Destination	–	*	*	0	0
EXG	Exchange Register	Rx ⟷ Ry	–	–	–	–	–
EXT	Sign Extend	(Destination) Sign-extended → Destination	–	*	*	0	0
JMP	Jump	Destination → PC	–	–	–	–	–
JSR	Jump to Subroutine	PC → – (SP); Destination → PC	–	–	–	–	–
LEA	Load Effective Address	Destination → An	–	–	–	–	–
LINK	Link and Allocate	An → – (SP); SP → An; SP + displacement → SP	–	–	–	–	–
LSL, LSR	Logical Shift	(Destination) Shifted by <count> → Destination	*	*	*	0	*
MOVE	Move Data from Source to Destination	(Source) → Destination	–	*	*	0	0
MOVE to CCR	Move to Condition Code	(Source) → CCR	*	*	*	*	*
MOVE to SR	Move to the Status Register	(Source) → SR	*	*	*	*	*
MOVE from SR	Move from the Status Register	SR → Destination	–	–	–	–	–
MOVE USP	Move User Stack Pointer	USP → An; An → USP	–	–	–	–	–
MOVEA	Move Address	(Source) → Destination	–	–	–	–	–
MOVEM	Move Multiple Registers	Registers → Destination (Source) → Registers	–	–	–	–	–
MOVEP	Move Peripheral Data	(Source) → Destination	–	–	–	–	–

Table 7-2. Instruction Sheet (Sheet 2 of 2)

Mnemonic	Description	Operation	Condition Codes				
			X	N	Z	V	C
MOVEQ	Move Quick	Immediate Data → Destination	–	*	*	0	0
MULS	Signed Multiply	(Destination) × (Source) → Destination	–	*	*	0	0
MULU	Unsigned Multiply	(Destination) x (Source) → Destination	–	*	*	0	0
NBCD	Negate Decimal with Extend	$0 - (\text{Destination})_{10} - x \rightarrow$ Destination	*	U	*	U	*
NEG	Negate	0 – (Destination) → Destination	*	*	*	*	*
NEGX	Negate with Extend	0 – (Destination) – x → Destination	*	*	*	*	*
NOP	No Operation	–	–	–	–	–	–
NOT	Logical Complement	~ (Destination) → Destination	–	*	*	0	0
OR	Inclusive OR Logical	(Destination) v (Source) → Destination	–	*	*	0	0
ORI	Inclusive OR Immediate	(Destination) v Immediate Data → Destination	–	*	*	0	0
PEA	Push Effective Address	Destination → – (SP)	–	–	–	–	–
RESET	Reset External Devices	–	–	–	–	–	–
ROL, ROR	Rotate (Without Extend)	(Destination) Rotated by <count> → Destination	–	*	*	0	*
ROXL, ROXR	Rotate with Extend	(Destination) Rotated by <count> → Destination	*	*	*	0	*
RTE	Return from Exception	(SP) + → SR; (SP) + → PC	*	*	*	*	*
RTR	Return and Restore Condition Codes	(SP) + → CC; (SP) + → PC	*	*	*	*	*
RTS	Return from Subroutine	(SP) + → PC	–	–	–	–	–
SBCD	Subtract Decimal with Extend	$(\text{Destination})_{10} - (\text{Source})_{10} - x \rightarrow$ Destination	*	U	*	U	*
S_{CC}	Set According to Condition	If $_{CC}$ then 1's → Destination else 0's → Destination	–	–	–	–	–
STOP	Load Status Register and Stop	Immediate Data → SR; STOP	*	*	*	*	*
SUB	Subtract Binary	(Destination) – (Source) → Destination	*	*	*	*	*
SUBA	Subtract Address	(Destination) – (Source) → Destination	–	–	–	–	–
SUBI	Subtract Immediate	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBQ	Subtract Quick	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBX	Subtract with Extend	(Destination) – (Source) – x → Destination	*	*	*	*	*
SWAP	Swap Register Halves	Register [31:16] ⟷ Register [15:0]	–	*	*	0	0
TAS	Test and Set an Operand	(Destination) Tested → CC; 1 → [7] OF Destination	–	*	*	0	0
TRAP	Trap	PC → – (SSP); SR → – (SSP); (Vector) → PC	–	–	–	–	–
TRAPV	Trap on Overflow	If V then TRAP	–	–	–	–	–
TST	Test an Operand	(Destination) Tested → CC	–	*	*	0	0
UNLK	Unlink	An → SP; (SP) + → An	–	–	–	–	–

● logical exclusive OR
Λ logical AND
v logical OR
~ logical complement

* affected
– unaffected
0 cleared
1 set
U undefined

7.1.2 Instruction Prefetch

The TS68008 uses a two-word tightly-coupled instruction prefetch mechanism to enhance performance. This mechanism is described in terms of the microcode operations involved. If the execution of an instruction is defined to begin when the microroutine for that instruction is entered, some features of the prefetch mechanism can be described.

1) When execution of an instruction begins, the operation word and the word following have already been fetched. The operation word is in the instruction decoder.

2) In the case of multiword instructions, as each additional word of the instruction is used internally, a fetch is made to the instruction stream to replace it.
3) The last fetch from the instruction stream is made when the operation word is discarded and decoding is started on the next instruction.
4) If the instruction is a single-word instruction causing a branch, the second word is not used. But because this word is fetched by the preceding instruction, it is impossible to avoid this superfluous fetch. In the case of an interrupt or trace exception, neither word is used.
5) The program counter usually points to the last word fetched from the instruction stream.

7.2 INSTRUCTION EXECUTION TIMES

The following paragraphs contain listings of the instruction execution times in terms of external clock (CLK) periods. In this timing data, it is assumed that both memory read and write cycle times are four clock periods. Any wait states caused by a longer memory cycle must be added to the total instruction time. The number of bus read and write cycles for each instruction is also included with the timing data. This data is enclosed in parenthesis following the execution periods and is shown as: (r/w) where r is the number of read cycles and w is the number of write cycles. **The number of periods includes instruction fetch and all applicable operand fetches and stores.**

7.2.1 Operand Effective Address Calculation Times

Table 7-3 lists the number of clock periods required to compute an instruction's effective address. It includes fetching of any extension words, the address computation, and fetching of the memory operand. The number of bus read and write cycles is shown in parenthesis as (r/w). Note there are no write cycles involved in processing the effective address.

Table 7-3. Effective Address Calculation Times

Addressing Mode		Byte	Word	Long
	Register			
Dn	Data Register Direct	**0**(0/0)	**0**(0/0)	**0**(0/0)
An	Address Register Direct	**0**(0/0)	**0**(0/0)	**0**(0/0)
	Memory			
(An)	Address Register Indirect	**4**(1/0)	**8**(2/0)	**16**(4/0)
(An) +	Address Register Indirect with Postincrement	**4**(1/0)	**8**(2/0)	**16**(4/0)
– (An)	Address Register Indirect with Predecrement	**6**(1/0)	**10**(2/0)	**18**(4/0)
d(An)	Address Register Indirect with Displacement	**12**(3/0)	**16**(4/0)	**24**(6/0)
d(An, ix)*	Address Register Indirect with Index	**14**(3/0)	**18**(4/0)	**26**(6/0)
xxx.W	Absolute Short	**12**(3/0)	**16**(4/0)	**24**(6/0)
xxx.L	Absolute Long	**20**(5/0)	**24**(6/0)	**32**(8/0)
d(PC)	Program Counter with Displacement	**12**(3/0)	**16**(4/0)	**24**(6/0)
d(PC, ix)	Program Counter with Index	**14**(3/0)	**18**(4/0)	**26**(6/0)
#xxx	Immediate	**8**(2/0)	**8**(2/0)	**16**(4/0)

* The size of the index register (ix) does not affect execution time.

7.2.2 Move Instruction Execution Times

Tables 7-4, 7-5, and 7-6 indicate the number of clock periods for the move instruction. This data includes instruction fetch, operand reads, and operand writes. The number of bus read and write cycles is shown in parenthesis as: (r/w).

Table 7-4. Move Byte Instruction Execution Times

Source	Destination								
	Dn	An	(An)	(An)+	-(An)	d(An)	d(An, x)*	xxx.W	xxx.L
Dn	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	20(4/1)	22(4/1)	20(4/1)	28(6/1)
An	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	20(4/1)	22(4/1)	20(4/1)	28(6/1)
(An)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	24(5/1)	26(5/1)	24(5/1)	32(7/1)
(An)+	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	24(5/1)	26(5/1)	24(5/1)	32(7/1)
-(An)	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	26(5/1)	28(5/1)	26(5/1)	34(7/1)
d(An)	20(5/0)	20(5/0)	24(5/1)	24(5/1)	24(5/1)	32(7/1)	34(7/1)	32(7/1)	40(9/1)
d(An, ix)*	22(5/0)	22(5/0)	26(5/1)	26(5/1)	26(5/1)	34(7/1)	36(7/1)	34(7/1)	42(9/1)
xxx.W	20(5/0)	20(5/0)	24(5/1)	24(5/1)	24(5/1)	32(7/1)	34(7/1)	32(7/1)	40(9/1)
xxx.L	28(7/0)	28(7/0)	32(7/1)	32(7/1)	32(7/1)	40(9/1)	42(9/1)	40(9/1)	48(11/1)
d(PC)	20(5/0)	20(5/0)	24(5/1)	24(5/1)	24(5/1)	32(7/1)	34(7/1)	32(7/1)	40(9/1)
d(PC, ix)*	22(5/0)	22(5/0)	26(5/1)	26(5/1)	26(5/1)	34(7/1)	36(7/1)	34(7/1)	42(9/1)
#xxx	16(4/0)	16(4/0)	20(4/1)	20(4/1)	20(4/1)	28(6/1)	30(6/1)	28(6/1)	36(8/1)

* The size of the index register (ix) does not affect execution time.

Table 7-5. Move Word Instruction Execution Times

Source	Destination								
	Dn	An	(An)	(An)+	-(An)	d(An)	d(An, ix)*	xxx.W	xxx.L
Dn	8(2/0)	8(2/0)	16(2/2)	16(2/2)	16(2/2)	24(4/2)	26(4/2)	20(4/2)	32(6/2)
An	8(2/0)	8(2/0)	16(2/2)	16(2/2)	16(2/2)	24(4/2)	26(4/2)	20(4/2)	32(6/2)
(An)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	32(6/2)	34(6/2)	32(6/2)	40(8/2)
(An)+	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	32(6/2)	34(6/2)	32(6/2)	40(8/2)
-(An)	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	34(6/2)	32(6/2)	34(6/2)	42(8/2)
d(An)	24(6/0)	24(6/0)	32(6/2)	32(6/2)	32(6/2)	40(8/2)	42(8/2)	40(8/2)	48(10/2)
d(An, ix)*	26(6/0)	26(6/0)	34(6/2)	34(6/2)	34(6/2)	42(8/2)	44(8/2)	42(8/2)	50(10/2)
xxx.W	24(6/0)	24(6/0)	32(6/2)	32(6/2)	32(6/2)	40(8/2)	42(8/2)	40(8/2)	48(10/2)
xxx.L	32(8/0)	32(8/0)	40(8/2)	40(8/2)	40(8/2)	48(10/2)	50(10/2)	48(10/2)	56(12/2)
d(PC)	24(6/0)	24(6/0)	32(6/2)	32(6/2)	32(6/2)	40(8/2)	42(8/2)	40(8/2)	48(10/2)
d(PC, ix)*	26(6/0)	26(6/0)	34(6/2)	34(6/2)	34(6/2)	42(8/2)	44(8/2)	42(8/2)	50(10/2)
#xxx	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	32(6/2)	34(6/2)	32(6/2)	40(8/2)

* The size of the index register (ix) does not affect execution time.

Table 7-6. Move Long Instruction Execution Times

Source	Destination								
	Dn	An	(An)	(An)+	-(An)	d(An)	d(An, ix)*	xxx.W	xxx.L
Dn	8(2/0)	8(2/0)	24(2/4)	24(2/4)	24(2/4)	32(4/4)	34(4/4)	32(4/4)	40(6/4)
An	8(2/0)	8(2/0)	24(2/4)	24(2/4)	24(2/4)	32(4/4)	34(4/4)	32(4/4)	40(6/4)
(An)	24(6/0)	24(6/0)	40(6/4)	40(6/4)	40(6/4)	48(8/4)	50(8/4)	48(8/4)	56(10/4)
(An)+	24(6/0)	24(6/0)	40(6/4)	40(6/4)	40(6/4)	48(8/4)	50(8/4)	48(8/4)	56(10/4)
-(An)	26(6/0)	26(6/0)	42(6/4)	42(6/4)	42(6/4)	50(8/4)	52(8/4)	50(8/4)	58(10/4)
d(An)	32(8/0)	32(8/0)	48(8/4)	48(8/4)	48(8/4)	56(10/4)	58(10/4)	56(10/4)	64(12/4)
d(An, ix)*	34(8/0)	34(8/0)	50(8/4)	50(8/4)	50(8/4)	58(10/4)	60(10/4)	58(10/4)	66(12/4)
xxx.W	32(8/0)	32(8/0)	48(8/4)	48(8/4)	48(8/4)	56(10/4)	58(10/4)	56(10/4)	64(12/4)
xxx.L	40(10/0)	40(10/0)	56(10/4)	56(10/4)	56(10/4)	64(12/4)	66(12/4)	64(12/4)	72(14/4)
d(PC)	32(8/0)	32(8/0)	48(8/4)	48(8/4)	48(8/4)	56(10/4)	58(10/4)	56(10/4)	64(12/4)
d(PC, ix)*	34(8/0)	34(8/0)	50(8/4)	50(8/4)	50(8/4)	58(10/4)	60(10/4)	58(10/4)	66(12/4)
#xxx	24(6/0)	24(6/0)	40(6/4)	40(6/4)	40(6/4)	48(8/4)	50(8/4)	48(8/4)	56(10/4)

* The size of the index register (ix) does not affect execution time.

7.2.3 Standard Instruction Execution Times

The number of clock periods shown in Table 7-7 indicates the time required to perform the operations, store the results, and read the next instruction. The number of bus read and write cycles is shown in parenthesis as: (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated. In Table 7-7 the headings have the following meanings: An = address register operand, Dn = data register operand, ea = an operand specified by an effective address, and M = memory effective address operand.

Table 7-7. Standard Instruction Execution Times

Instruction	Size	op <ea>, An	op <ea>, Dn	op Dn, <M>
ADD	Byte	–	8(2/0) +	12(2/1) +
	Word	12(2/0) +	8(2/0) +	16(2/2) +
	Long	10(2/0) + **	10(2/0) + **	24(2/4) +
AND	Byte	–	8(2/0) +	12(2/1) +
	Word	–	8(2/0) +	16(2/2) +
	Long	–	10(2/0) + **	24(2/4) +
CMP	Byte	–	8(2/0) +	–
	Word	10(2/0) +	8(2/0) +	–
	Long	10(2/0) +	10(2/0) +	–
DIVS		–	162(2/0) + *	–
DIVU		–	144(2/0) + *	–
EOR	Byte	–	8(2/0) + ***	12(2/1) +
	Word	–	8(2/0) + ***	16(2/2) +
	Long	–	12(2/0) + ***	24(2/4) +
MULS		–	74(2/0) + *	–
MULU		–	74(2/0) + *	–
OR	Byte	–	8(2/0) +	12(2/1) +
	Word	–	8(2/0) +	16(2/2) +
	Long	–	10(2/0) + **	24(2/4) +
SUB	Byte	–	8(2/0) +	12(2/1) +
	Word	12(2/0) +	8(2/0) +	16(2/2) +
	Long	10(2/0) + **	10(2/0) + **	24(2/4) +

NOTES:

\+ Add effective address calculation time

* Indicates maximum value

** The base time of 10 clock periods is increased to 12 if the effective address mode is register direct or immediate (effective address time should also be added).

*** Only available effective address mode is data register direct

DIVS, DIVU – The divide algorithm used by the TS68008 provides less than 10% difference between the best and worst case timings.

MULS, MULU – The multiply algorithm requires 42 + 2n clocks where n is defined as:

MULS: n = tag the <ea> with a zero as the MSB; n is the resultant number of 10 or 01 patterns in the 17-bit source, i.e., worst case happens when the source is $5555.

MULU: n = the number of ones in the <ea>

7.2.4 Immediate Instruction Execution Times

The number of clock periods shown in Table 7-8 includes the time to fetch immediate operands, perform the operations, store the results, and read the next operation. The number of bus read and write cycles is shown in parenthesis as: (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated. In Table 7-8, the headings have the following meanings: # = immediate operand, Dn = data register operand, An = address register operand, and M = memory operand.

Table 7-8. Immediate Instruction Clock Periods

Instruction	Size	op#, Dn	op#,An	op#, M
ADDI	Byte	16(4/0)	–	20(4/1)+
	Word	16(4/0)	–	24(4/2)+
	Long	28(6/0)	–	40(6/4)+
ADDQ	Byte	8(2/0)	–	12(2/1)+
	Word	8(2/0)	12(2/0)	16(2/2)+
	Long	12(2/0)	12(2/0)	24(2/4)+
ANDI	Byte	16(4/0)	–	20(4/1)+
	Word	16(4/0)	–	24(4/2)+
	Long	28(6/0)	–	40(6/4)+
CMPI	Byte	16(4/0)	–	16(4/0)+
	Word	16(4/0)	–	16(4/0)+
	Long	26(6/0)	–	24(6/0)+
EORI	Byte	16(4/0)	–	20(4/1)+
	Word	16(4/0)	–	24(4/2)+
	Long	28(6/0)	–	40(6/4)+
MOVEQ	Long	8(2/0)	–	–
ORI	Byte	16(4/0)	–	20(4/1)+
	Word	16(4/0)	–	24(4/2)+
	Long	28(6/0)	–	40(6/4)+
SUBI	Byte	16(4/0)	–	12(2/1)+
	Word	16(4/0)	–	16(2/2)+
	Long	28(6/0)	–	24(2/4)+
SUBQ	Byte	8(2/0)	–	20(4/1)+
	Word	8(2/0)	12(2/0)	24(4/2)+
	Long	12(2/0)	12(2/0)	40(6/4)+

+ add effective address calculation time

7.2.5 Single Operand Instruction Execution Times

Table 7-9 indicates the number of clock periods for the single operand instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 7-9. Single Operand Instruction Execution Times

Instruction	Size	Register	Memory
CLR	Byte	8(2/0)	12(2/1)+
	Word	8(2/0)	16(2/2)+
	Long	10(2/0)	24(2/4)+
NBCD	Byte	10(2/0)	12(2/1)+
NEG	Byte	8(2/0)	12(2/1)+
	Word	8(2/0)	16(2/2)+
	Long	10(2/0)	24(2/4)+
NEGX	Byte	8(2/0)	12(2/1)+
	Word	8(2/0)	16(2/2)+
	Long	10(2/0)	24(2/4)+
NOT	Byte	8(2/0)	12(2/1)+
	Word	8(2/0)	16(2/2)+
	Long	10(2/0)	24(2/4)+
Scc	Byte, False	8(2/0)	12(2/1)+
	Byte, True	10(2/0)	12(2/1)+
TAS	Byte	8(2/0)	14(2/1)+
TST	Byte	8(2/0)	8(2/0)+
	Word	8(2/0)	8(2/0)+
	Long	8(2/0)	8(2/0)+

+ add effective address calculation time.

7.2.6 Shift/Rotate Instruction Execution Times

Table 7-10 indicates the number of clock periods for the shift and rotate instructions. The number of bus read and write cycles is shown in parenthesis as: (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 7-10. Shift/Rotate Instruction Clock Periods

Instruction	Size	Register	Memory
ASR, ASL	Byte	10 + 2n(2/0)	–
	Word	10 + 2n(2/0)	16(2/2) +
	Long	12 + 2n(2/0)	–
LSR, LSL	Byte	10 + 2n(2/0)	–
	Word	10 + 2n(2/0)	16(2/2) +
	Long	12 + 2n(2/0)	–
ROR, ROL	Byte	10 + 2n(2/0)	–
	Word	10 + 2n(2/0)	16(2/2) +
	Long	12 + 2n(2/0)	–
ROXR, ROXL	Byte	10 + 2n(2/0)	–
	Word	10 + 2n(2/0)	16(2/2) +
	Long	12 + 2n(2/0)	–

+ add effective address calculation time
n is the shift count

7.2.7 Bit Manipulation Instruction Execution Times

Table 7-11 indicates the number of clock periods required for the bit manipulation instructions. The number of bus read and write cycles is shown in parenthesis as: (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 7-11. Bit Manipulation Instruction Execution Times

Instruction	Size	Dynamic		Static	
		Register	Memory	Register	Memory
BCHG	Byte	–	12(2/1) +	–	20(4/1) +
	Long	12(2/0)*	–	20(4/0)*	–
BCLR	Byte	–	12(2/1) +	–	20(4/1) +
	Long	14(2/0)*	–	22(4/0)*	–
BSET	Byte	–	12(2/1) +	–	20(4/1) +
	Long	12(2/0)*	–	20(4/0)*	–
BTST	Byte	–	8(2/0) +	–	16(4/0) +
	Long	10(2/0)	–	18(4/0)	–

+ add effective address calculation time
* indicates maximum value

7.2.8 Conditional Instruction Execution Times

Table 7-12 indicates the number of clock periods required for the conditional instructions. The number of bus read and write cycles is indicated in parenthesis as: (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 7-12. Conditional Instruction Execution Times

Instruction	Displacement	Trap or Branch Taken	Trap or Branch Not Taken
Bcc	Byte	18(4/0)	12(2/0)
	Word	18(4/0)	20(4/0)
BRA	Byte	18(4/0)	–
	Word	18(4/0)	–
BSR	Byte	34(4/4)	–
	Word	34(4/4)	–
DBcc	CC True	–	20(4/0)
	CC False	18(4/0)	26(6/0)
CHK	–	68(8/6) + *	14(2/0) +
TRAP	–	62(8/6)	–
TRAPV	–	66(10/6)	8(2/0)

+ add effective address calculation time
* indicates maximum value

7.2.9 JMP, JSR, LEA, PEA, and MOVEM Instruction Execution Times

Table 7-13 indicates the number of clock periods required for the jump, jump-to-subroutine, load effective address, push effective address, and move multiple registers instructions. The number of bus read and write cycles is shown in parenthesis as: (r/w).

Table 7-13. JMP, JSR, LEA, PEA, and MOVEM Instruction Execution Times

Instruction	Size	(An)	(An) +	– (An)	d(An)	d(An, ix)*	xxx.W	xxx.L	d(PC)	d(PC,ix)*
JMP	–	16(4/0)	–	–	18(4/0)	22(4/0)	18(4/0)	24(6/0)	18(4/0)	22(4/0)
JSR	–	32(4/4)	–	–	34(4/4)	38(4/4)	34(4/4)	40(6/4)	34(4/4)	38(4/4)
LEA	–	8(2/0)	–	–	16(4/0)	20(4/0)	16(4/0)	24(6/0)	16(4/0)	20(4/0)
PEA	–	24(2/4)	–	–	32(4/4)	36(4/4)	32(4/4)	40(6/4)	32(4/4)	36(4/4)
MOVEM M → R	Word	24 + 8n (6 + 2n/0)	24 + 8n (6 + 2n/0)	– –	32 + 8n (8 + 2n/0)	34 + 8n (8 + 2n/0)	32 + 8n (10 + n/0)	40 + 8n (10 + 2n/0)	32 + 8n (8 + 2n/0)	34 + 8n (8 + 2n/0)
	Long	24 + 16n (6 + 4n/0)	24 + 16n (6 + 4n/0)	– –	32 + 16n (8 + 4n/0)	34 + 16n (8 + 4n/0)	32 + 16n (8 + 4n/0)	40 + 16n (8 + 4n/0)	32 + 16n (8 + 4n/0)	34 + 16n (8 + 4n/0)
MOVEM R → M	Word	16 + 8n (4/2n)		16 + 8n (4/2n)	24 + 8n (6/2n)	26 + 8n (6/2n)	24 + 8n (6/2n)	32 + 8n (8/2n)		
	Long	16 + 16n (4/4n)		16 + 16n (4/4n)	24 + 16n (6/4n)	26 + 16n (6/4n)	24 + 16n (8/4n)	32 + 16n (6/4n)		

n is the number of registers to move
* is the size of the index register (ix) does not affect the instruction's execution time

7.2.10 Multi-Precision Instruction Execution Times

Table 7-14 indicates the number of clock periods for the multi-precision instructions. The number of clock periods includes the time to fetch both operands, perform the operations, store the results, and read the next instructions. The number of read and write cycles is shown in parenthesis as: (r/w).

In Table 7-14, the headings have the following meanings: Dn = data register operand and M = memory operand.

Table 7-14. Multi-Precision Instruction Execution Times

Instruction	Size	op Dn, Dn	op M, M
ADDX	Byte	8(2/0)	22(4/1)
	Word	8(2/0)	50(6/2)
	Long	12(2/0)	58(10/4)
CMPM	Byte	–	16(4/0)
	Word	–	24(6/0)
	Long	–	40(10/0)
SUBX	Byte	8(2/0)	22(4/1)
	Word	8(2/0)	50(6/2)
	Long	12(2/0)	58(10/4)
ABCD	Byte	10(2/0)	20(4/1)
SBCD	Byte	10(2/0)	20(4/1)

7.2.11 Miscellaneous Instruction Execution Times

Tables 7-15 and 7-16 indicate the number of clock periods for the following miscellaneous instructions. The number of bus read and write cycles is shown in parenthesis as: (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

7.2.12 Exception Processing Execution Times

Table 7-17 indicates the number of clock periods for exception processing. The number of clock periods includes the time for all stacking, the vector fetch, and the fetch of the first instruction of the handler routine. The number of bus read and write cycles is shown in parenthesis as: (r/w).

Table 7-15. Miscellaneous Instruction Execution Times

Instruction	Register	Memory
ANDI to CCR	32(6/0)	
ANDI to SR	32(6/0)	
EORI to CCR	32(6/0)	
EORI to SR	32(6/0)	
EXG	10(2/0)	
EXT	8(2/0)	
LINK	32(4/4)	
MOVE to CCR	18(4/0)	18(4/0)+
MOVE to SR	18(4/0)	18(4/0)+
MOVE from SR	10(2/0)	16(2/2)+
MOVE to USP	8(2/0)	
MOVE from USP	8(2/0)	
NOP	8(2/0)	
ORI to CCR	32(6/0)	
ORI to SR	32(6/0)	
RESET	136(2/0)	
RTE	40(10/0)	
RTR	40(10/0)	
RTS	32(8/0)	
STOP	4(0/0)	
SWAP	8(2/0)	
UNLK	24(6/0)	

+ add effective address calculation time

Table 7-16. Move Peripheral Instruction Execution Times

Instruction	Size	Register → Memory	Memory → Register
MOVEP	Word	24(4/2)	24(6/0)
	Long	32(4/4)	32(8/0)

+ add effective address calculation time

Table 7-17. Exception Processing Execution Times

Exception	Periods
Address Error	94(8/14)
Bus Error	94(8/14)
CHK Instruction	68(8/6)+
Interrupt	72(9/16)*
Illegal Instruction	62(8/6)
Privileged Instruction	62(8/6)
Trace	62(8/6)
TRAP Instruction	62(8/6)
TRAPV Instruction	66(10/6)
Divide by Zero	66(8/6)+
$\overline{\text{RESET}}$**	64(12/0)

+ add effective address calculation time

* The interrupt acknowledge bus cycle is assumed to take four external clock periods.

** Indicates the time from when $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are first sampled as negated to when instruction execution starts.

SECTION 8
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the MC68008.

8.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range TS68008C TS68008V	T_A	0 to 70 0 to 70 −40 to 85	°C
Storage Temperature	T_{stg}	−55 to 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either ground or V_{CC}).

8.2 THERMAL CHARACTERISTICS

Characteristic	Value		Rating
	θ_{JA}	θ_{JC}	
Thermal Resistance			
Ceramic DIL	40	15*	°C/W
Plastic DIL	40	20*	
PLCC	50	30*	

*Estimated

8.3 POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts − Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins − User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

The curve shown in Figure 8-1 gives the graphic solution to these equations for the specification power dissipation of 1.50 watts over the ambient temperature range of $-55°C$ to $125°C$ using a θ_{JA} of 45°C/W, a typical value for packages specified.

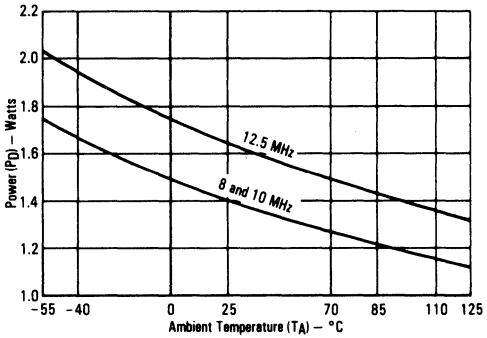


Figure 8-1. TS68008 Power Dissipation (P_D) vs Ambient Temperature (T_A)

The total thermal resistance of a package (θ_{JA}) can be separated into two components, θ_{JC} and θ_{CA}, representing the barrier to heat flow from the semiconductor junction to the package (case) surface (θ_{JC}) and from the case to the outside ambient (θ_{CA}). These terms are related by the equation:

$$\theta_{JA} = \theta_{JC} + \theta_{CA} \quad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling and thermal convention. Thus good thermal management on the part of the user can significantly reduce θ_{CA} so that $\theta_{JA} = \theta_{JC}$. Substitution of θ_{JC} for θ_{JA} in equation 1 will result in a lower semiconductor junction temperature.

8.4 DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 Vdc ±5%; GND = 0 Vdc; T_A = 0°C to 70°C; see Figures 8-2, 8-3, and 8-4)

Characteristic		Symbol	Min	Max	Unit
Input High Voltage		V_{IH}	2.0	V_{CC}	V
Input Low Voltage		V_{IL}	GND – 0.3	0.8	V
Input Leakage Current @ 5.25 V $\overline{BERR}$, $\overline{BR}$, $\overline{DTACK}$, CLK, $\overline{IPL0/2}$, $\overline{IPL1}$, $\overline{VPA}$, $\overline{HALT}$, $\overline{RESET}$, $\overline{BGACK}$		I_{in}	–	20	μA
Hi-Z (Off State) Input Current @ 2.4 V/0.4 V A0-A19, $\overline{AS}$, D0-D7, FC0-FC2, $\overline{DS}$, R/$\overline{W}$		I_{TSI}	–	20	μA
Output High Voltage (I_{OH} = – 400 μA)	E, A0-A19, AS, BG, D0-D7, FC0-FC2, $\overline{DS}$, R/$\overline{W}$, $\overline{VMA}$	V_{OH}	2.4	–	V
Output Low Voltage					
(I_{OL} = 1.6 mA)	$\overline{HALT}$		–	0.5	
(I_{OL} = 3.2 mA)	A0-A19, $\overline{BG}$, FC0-FC2	V_{OL}	–	0.5	V
(I_{OL} = 5.0 mA)	$\overline{RESET}$		–	0.5	
(I_{OL} = 5.3 mA)	E, $\overline{AS}$, D0-D7, $\overline{DS}$, R/$\overline{W}$		–	0.5	
Power Dissipation,* T_A = 0°C		P_D	–	1.5	W
Capacitance (V_{in} = 0 V, T_A = 25°C, Frequency = 1 MHz)**		C_{in}	–	20.0	pF

*During normal operation instantaneous V_{CC} current requirements may be as high as 1.5 A

**Capacitance is periodically sampled rather than 100% tested.

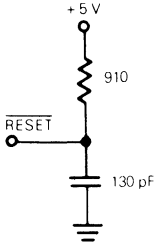


Figure 8-2. $\overline{RESET}$ Test Load

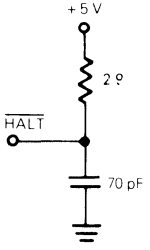


Figure 8-3. $\overline{HALT}$ Test Load

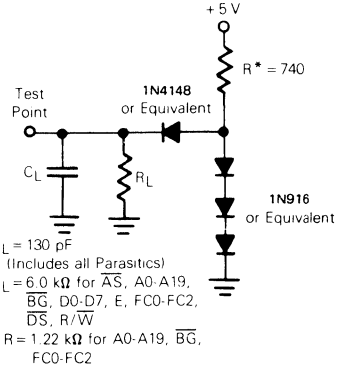


Figure 8-4. Test Loads

8.5 CLOCK TIMING (See Figure 8-5)

Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f	2.0	8.0	2.0	10.0	4.0	12.5	MHz
Cycle Time	t_{cyc}	125	500	100	500	80	250	ns
Clock Pulse Width	t_{CL}	55	250	45	250	35	125	ns
	t_{CH}	55	250	45	250	35	125	
Rise and Fall Times	t_{Cr}	–	10	–	10	–	5	ns
	t_{Cf}	–	10	–	10	–	5	

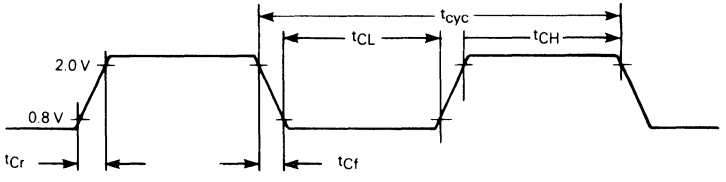


Figure 8-5. Input Clock Waveform

8.6 AC ELECTRICAL SPECIFICATIONS — READ CYCLES

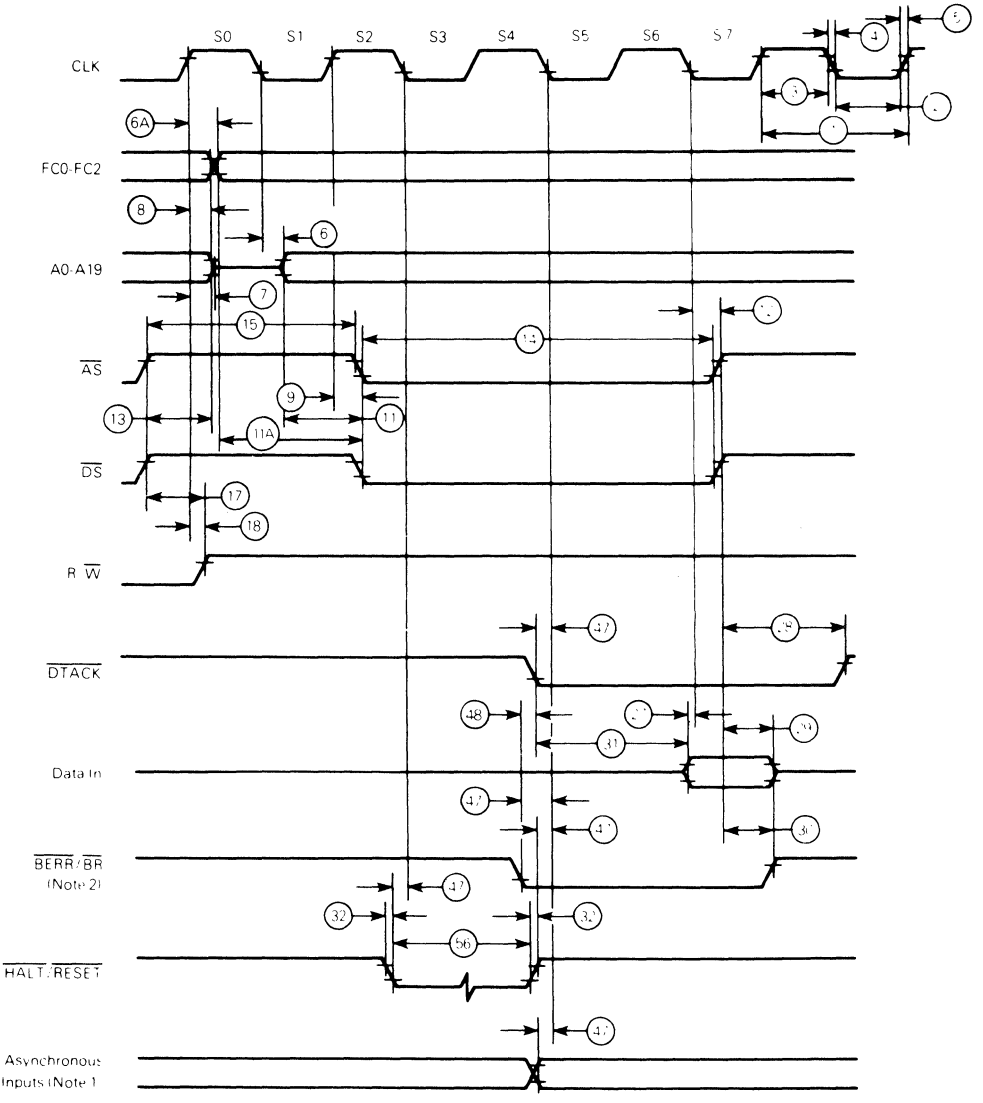
(V_{CC}=5.0 Vdc ± 5 % ; GND=0 Vdc ; T_A=T_L to T_H ; see Figure 8-6)

Num.	Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Clock Period	tCYC	125	500	100	500	80	250	ns
2	Clock Width Low	tCL	55	250	45	250	35	125	ns
3	Clock Width High	tCH	55	250	45	250	35	125	ns
4	Clock Fall Time	tCt	–	10	–	10	–	5	ns
5	Clock Rise Time	tCr	–	10	–	10	–	5	ns
6	Clock Low to Address Valid	tCLAV	–	70	–	60	–	55	ns
6A	Clock High to FC Valid	tCHFCV	–	70	–	60	–	55	Ns
7	Clock High to Address, Data Bus High Impedance (Maximum)	tCHADZ	–	80	–	70	–	60	ns
8	Clock High to Address, FC Invalid (Minimum)	tCHAFI	0	–	0	–	0	–	ns
9[1]	Clock High to $\overline{AS}$, $\overline{DS}$ Low	tCHSL	0	60	0	55	0	55	ns
11[2]	Address Valid to $\overline{AS}$, $\overline{DS}$ Low	tAVSL	30	–	20	–	0	–	ns
11A[2,6]	FC Valid to $\overline{AS}$, $\overline{DS}$ Low	tFCVSL	60	–	50	–	40	–	ns
12[1]	Clock Low to $\overline{AS}$, $\overline{DS}$ High	tCLSH	–	35	–	35	–	35	ns
13[2]	$\overline{AS}$, $\overline{DS}$ High to Address/FC Invalid	tSHARI	30	–	20	–	10	–	ns
14[2,5]	$\overline{AS}$, $\overline{DS}$ Width Low	tSL	270	–	195	–	160	–	ns
15[2]	$\overline{AS}$, $\overline{DS}$ Width High	tSH	150	–	105	–	65	–	ns
17[2]	$\overline{AS}$, $\overline{DS}$ High to R/$\overline{W}$ High	tSHRH	40	–	20	–	10	–	ns
18[1]	Clock High to R/$\overline{W}$ High	tCHRH	0	40	0	40	0	40	ns
27[5]	Data In to Clock Low (Setup Time)	tDICL	15	–	10	–	10	–	ns
28[2,5]	$\overline{AS}$, $\overline{DS}$ High to $\overline{DTACK}$ High	tSHDAH	0	245	0	190	0	150	ns
29	$\overline{AS}$, $\overline{DS}$ High to Data In Invalid (Hold Time)	tSHDII	0	–	0	–	0	–	ns
30	$\overline{AS}$, $\overline{DS}$ High to $\overline{BERR}$ High	tSHBEH	0	–	0	–	0	–	ns
31[2,5]	$\overline{DTACK}$ Low to Data Valid (Asynchronous Setup Time on Read)	tDALDI	–	90	–	65	–	50	ns
32	$\overline{HALT}$ and $\overline{RESET}$ Input Transition Time	tRHr,f	0	200	0	200	0	200	ns
47[5]	Asynchronous Input Setup Time	tASI	10	–	10	–	10	–	ns
48[3]	$\overline{BERR}$ Low to $\overline{DTACK}$ Low	tBELDAL	20	–	20	–	20	–	ns
56[4]	$\overline{HALT}$ / $\overline{RESET}$ Pulse Width	tHRPW	10	–	10	–	10	–	Clk.Per.

NOTES :

1. For a loading capacitance of less than or equal to 50 picofarads, subtract 5 nanoseconds from the values given in these columns.
2. Actual value depends on clock period.
3. If 47 is satisfied for both $\overline{DTACK}$ and $\overline{BERR}$, 48 may be 0 nanoseconds.
4. For power up the MPU must be held in $\overline{RESET}$ state for 100 milliseconds to allow stabilization of on-chip circuitry. After the system is powered up, 56 refers to the minimum pulse width required to reset the system.
5. If the asynchronous setup time (47) requirements are satisfied, the $\overline{DTACK}$ low-to-data setup time (31) requirement can be ignored. The data must only satisfy the data-in to clock-low setup time (27) for the following cycle.
6. Setup time to guarantee recognition on next falling edge of clock.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals Refer to other functional descriptions and their related diagrams for device operation



NOTES

1 Setup time for the asynchronous inputs IPL0, IPL1, and VPA guarantees their recognition at the next falling edge of the clock
2 BR need fall at this time only in order to insure being recognized at the end of this bus cycle
3 Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted

Figure 8-6. Read Cycle Timing Diagram

8.6 AC ELECTRICAL SPECIFICATIONS — WRITE CYCLES

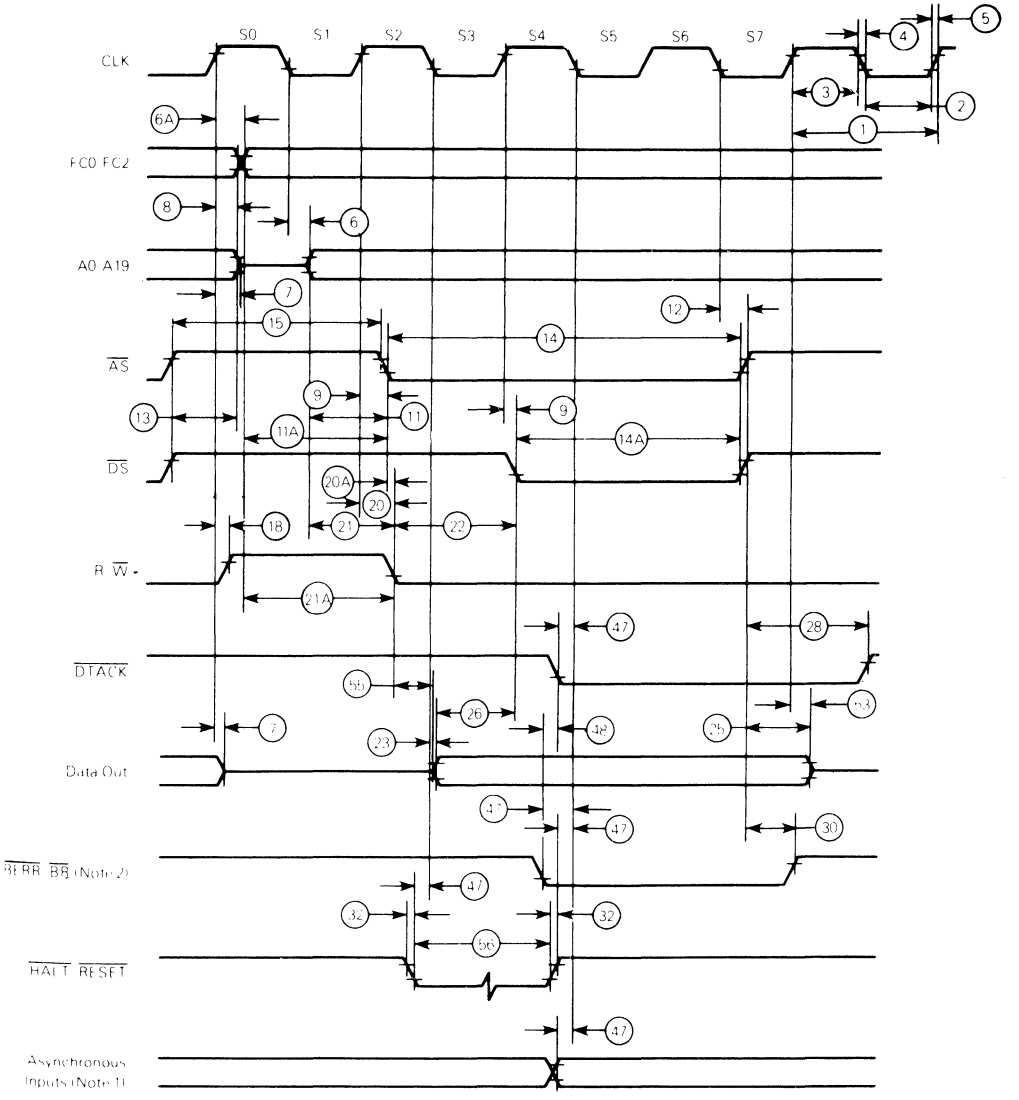
(V_{CC}=5.0 Vdc ± 5 % ; GND=0 Vdc ; T_A=T_L to T_H ; see Figure 8-7)

Num.	Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Clock Period	tCYC	125	500	100	500	80	250	ns
2	Clock Width Low	tCL	55	250	45	250	35	125	ns
3	Clock Width High	tCH	55	250	45	250	35	125	ns
4	Clock Fall Time	tCt	–	10	–	10	–	5	ns
5	Clock Rise Time	tCr	–	10	–	10	–	5	ns
6	Clock Low to Address Valid	tCLAV	–	70	–	60	–	55	ns
6A	Clock High to FC Valid	tCHFCV	–	70	–	60	–	55	Ns
7	Clock High to Address, Data Bus High Impedance (Maximum)	tCHADZ	–	80	–	70	–	60	ns
8	Clock High to Address, FC Invalid (Minimum)	tCHAFI	0	–	0	–	0	–	ns
9[1]	Clock High to $\overline{AS}$, $\overline{DS}$ Low	tCHSL	0	60	0	55	0	55	ns
11[2]	Address Valid to $\overline{AS}$ Low	tAVSL	30	–	20	–	0	–	ns
11A[2,7]	FC Valid to $\overline{AS}$ Low	tFCVSL	60	–	50	–	40	–	ns
12[1]	Clock Low to $\overline{AS}$, $\overline{DS}$ High	tCLSH	–	35	–	35	–	35	ns
13[2]	$\overline{AS}$, $\overline{DS}$ High to Address/FC Invalid	tSHARI	30	–	20	–	10	–	ns
14[2,5]	$\overline{AS}$ Low	tSL	270	–	195	–	160	–	ns
14A[2]	$\overline{DS}$ Width Low	tDSL	140	–	95	–	80	–	ns
15[2]	$\overline{AS}$, $\overline{DS}$ Width High	tSH	150	–	105	–	65	–	ns
18[1]	Clock High to R/$\overline{W}$ High	tCHRH	0	40	0	40	0	40	ns
20[1]	Clock High to R/$\overline{W}$ Low	tCHRL	–	40	–	40	–	40	ns
20A[6]	$\overline{AS}$, Low to R/$\overline{W}$ valid	tASRV	–	20	–	20	–	20	ns
21[2]	Address Valid to R/$\overline{W}$ Low	tAVRL	20	–	0	–	0	–	ns
21A[2,7]	FC Valid to R/$\overline{W}$ Low	tFCVRL	60	–	50	–	30	–	ns
22[2]	R/$\overline{W}$ Low to $\overline{DS}$ Low	tRLSL	80	–	50	–	30	–	ns
23	Clock Low to Data Out Valid	tCLDO	–	70	–	55	–	55	ns
25[2]	$\overline{AS}$, $\overline{DS}$ High to Data Out Invalid	tSHDOI	50	–	20	–	20	–	ns
26[2]	Data Out Valid to $\overline{DS}$ Low	tDOSL	35	–	20	–	20	–	ns
28[2,5]	$\overline{AS}$, $\overline{DS}$ High to $\overline{DTACK}$ High	tSHDAH	0	245	0	190	0	150	ns
30	$\overline{AS}$, $\overline{DS}$ High to $\overline{BERR}$ High	tSHBEH	0	–	0	–	0	–	ns
32	$\overline{HALT}$ and $\overline{RESET}$ Input Transition Time	tRHr,f	0	200	0	200	0	200	ns
47[5]	Asynchronous Input Setup Time	tASI	10	–	10	–	10	–	Ns
48[3]	$\overline{BERR}$ Low to $\overline{DTACK}$ Low	tBELDAL	20	–	20	–	20	–	ns
53	Clock High to Data Out Invalid	tCHDOI	0	–	0	–	0	–	ns
55	R/$\overline{W}$ to Data Bus Impedance Driven	tRLDBD	30	–	20	–	10	–	ns
56[4]	$\overline{HALT}$ / $\overline{RESET}$ Pulse Width	tHRPW	10	–	10	–	10	–	Clk.Per.

NOTES :

1. For a loading capacitance of less than or equal to 50 picofarads, subtract 5 nanoseconds from the values given in these columns.
2. Actual value depends on clock period.
3. If 47 is satisfied for both $\overline{DTACK}$ and $\overline{BERR}$, 48 may be 0 nanoseconds.
4. For power up the MPU must be held in $\overline{RESET}$ state for 100 milliseconds to allow stabilization of on-chip circuitry. After the system is powered up 56 refers to the minimum pulse width required to reset the system.
5. If the asynchronous setup time (47) requirements are satisfied, the $\overline{DTACK}$ low-to-data setup time (31) requirement can be ignored. The data must only satisfy the data-in to clock-low setup time (27) for the following cycle.
6. When $\overline{AS}$, and R/$\overline{W}$ are equally loaded (±20 %), subtract 10 nanoseconds from the values in these columns.
7. Setup time to guarantee recognition on next falling edge of clock.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.



NOTES

1. Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted.
2. Because of loading variations, R W may be valid after AS even though both are initiated by the rising edge of S2 (Specification 20A).

Figure 8-7. Write Cycle Timing Diagram

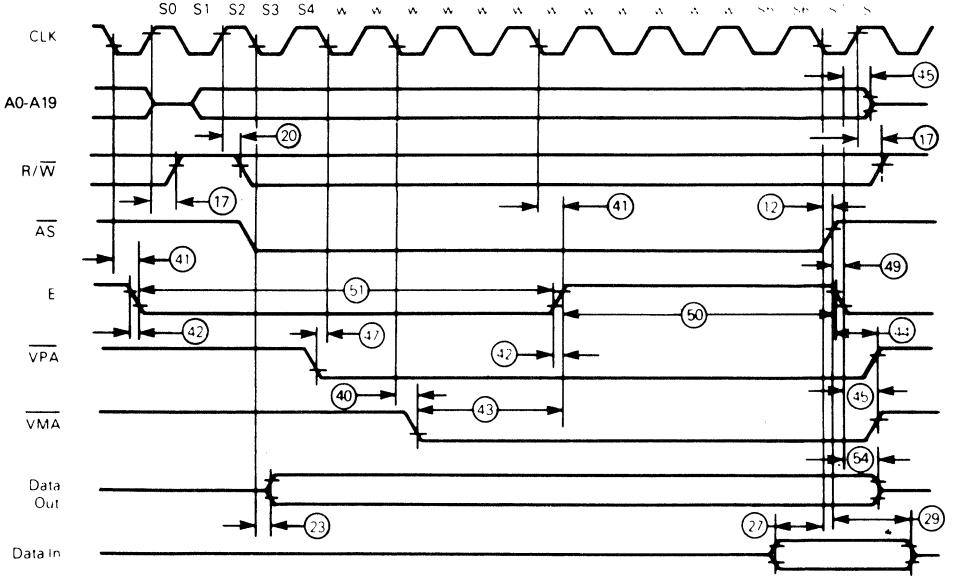
8.7 AC ELECTRICAL SPECIFICATIONS — TS 68008 to 6800 PERIPHERAL

(V_{CC}=5.0 Vdc ± 5 % ; GND=0 Vdc ; T_A=0° to 70°C ; see Figures 8-8 and 8-9)

Num.	Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		Unit
			Min	Max	Min	Max	Min	Max	
12[1]	Clock Low to $\overline{AS}$, $\overline{DS}$ High	tCLSH	–	35	–	35	–	35	ns
17[2]	$\overline{AS}$, $\overline{DS}$ High to R/$\overline{W}$ High (Read)	tSHRH	40	–	20	–	10	–	ns
18[1]	Clock High to R/$\overline{W}$ High	tCHRH	0	40	0	40	0	40	ns
20[1]	Clock High to R/$\overline{W}$ Low	tCHRL	–	40	–	40	–	40	ns
23	Clock Low to Data Out Valid (Write)	tCLDO	–	70	–	55	–	55	ns
27	Data In to Clock Low (Setup Time on Read)	tDICL	15	–	10	–	10	–	ns
29	$\overline{AS}$, $\overline{DS}$ High to Data In Invalid (Hold Time on Read)	tSHDII	0	–	0	–	0	–	ns
41	Clock Low to E Transition	tCLET	–	50	–	50	–	45	ns
42	E Output Rise and Fall Time	tEr,f	–	15	–	15	–	15	ns
44	$\overline{AS}$, $\overline{DS}$ High to $\overline{VPA}$ High	tSHVPH	0	120	0	90	0	70	ns
45	E Low to Control, Address Bus Invalid (Address Hold Time)	tELCAI	30	–	10	–	10	–	ns
47	Asynchronous Input Setup Time	tASI	10	–	10	–	10	–	ns
49[3]	$\overline{AS}$, $\overline{DS}$ High to E Low	tSHEL	-80	80	-80	80	-80	80	ns
50	E Width High	tEH	450	–	350	–	280	–	ns
51	E Width Low	tEL	700	–	550	–	440	–	ns
54	E Low to Data Out Invalid	tELDOI	30	–	20	–	15	–	ns

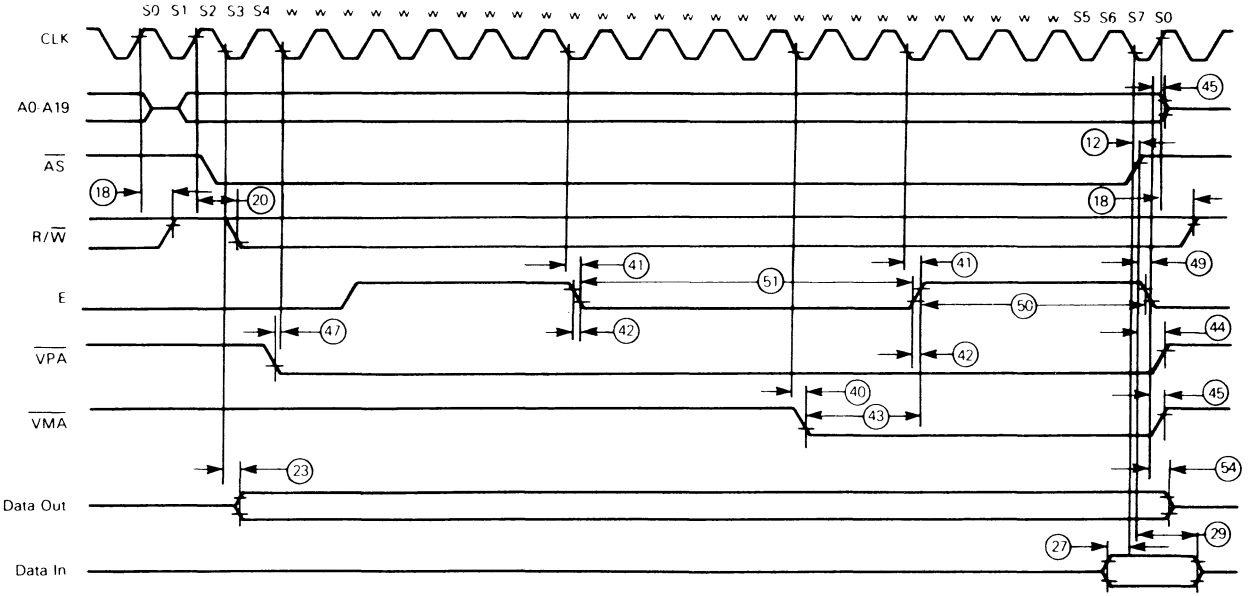
NOTES :

1. For a loading capacitance of less than or equal to 50 picofarads, subtract 5 nanoseconds from the values given in these columns.
2. Actual value depends on clock period.
3. The falling edge of S6 triggers both the negation of the strobes ($\overline{AS}$, and x $\overline{DS}$) and the falling edge of E. Either of these events can occur first, depending upon the loading on each signal. Specification 49 indicates the absolute maximum skew that will occur between the rising edge of the strobes and the falling edge of the E clock.



NOTE This timing diagram is included for those who wish to design their own circuit to generate $\overline{VMA}$. It shows the best case possibly attainable

Figure 8-8. TS68008 to 6800 Peripheral Timing Diagram — Best Case



NOTE: This timing diagram is included for those who wish to design their own circuit to generate $\overline{VMA}$. It shows the worst case possibly attainable.

Figure 8-9. TS68008 to 6800 Peripheral Timing Diagram — Worst Case

8.8 AC ELECTRICAL SPECIFICATIONS — BUS ARBITRATION

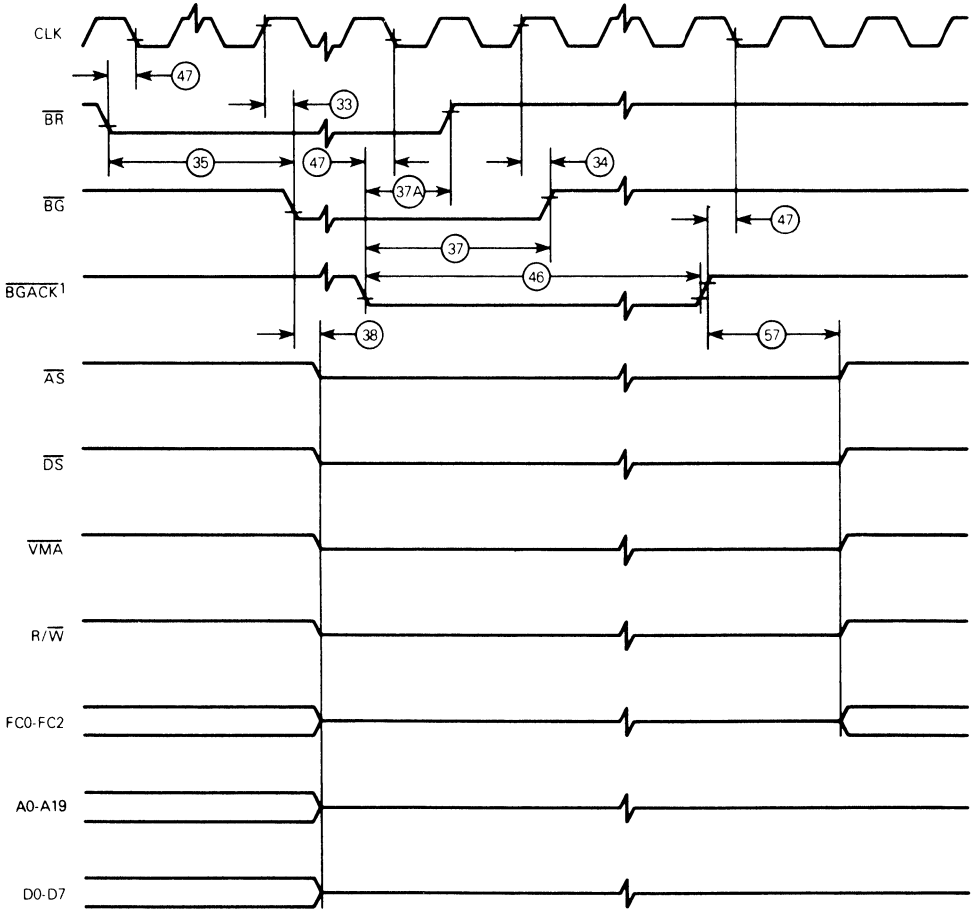
(V_{CC}=5.0 Vdc ± 5 % ; GND=0 Vdc ; $T_A = T_L$ to T_H ; see Figures 8-10, 8-11, and 8-12)

Num.	Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		Unit
			Min	Max	Min	Max	Min	Max	
7	Clock High to Address, Data Bus High Impedance	tCHADZ	–	80	–	70	–	60	ns
16	Clock High to Control Bus High Impedance	tCHCZ	–	80	–	70	–	60	ns
33	Clock High to $\overline{BG}$ Low	tCHGL	–	40	–	40	–	40	ns
34	Clock High to $\overline{BG}$ High	tCHGH	–	40	–	40	–	40	ns
35	$\overline{BR}$, Low to $\overline{BG}$ Low	tBRLGL	1.5	90 ns +3.5	1.5	80 ns +3.5	1.5	80 ns +3.5	Clk.Per.
36[1]	$\overline{BR}$ High to $\overline{BG}$ High	tBRHGH	1.5	90 ns +3.5	1.5	80 ns +3.5	1.5	80 ns +3.5	Clk.Per.
37	$\overline{BGACK}$ Low to $\overline{BG}$ High (52-Pin Version Only)	tGALGH	1.5	90 ns +3.5	1.5	80 ns +3.5	1.5	80 ns +3.5	Clk.Per.
37A[2]	$\overline{BGACK}$ Low to $\overline{BR}$ High (52-Pin Version Only)	tGALBRH	20	1.5 Clocks	20	1.5 Clocks	20	1.5 Clocks	ns
38	$\overline{BG}$ Low to Control, Address, Data Bus High Impedance ($\overline{AS}$ High)	tGLZ	–	80	–	70	–	60	ns
39	$\overline{BG}$ Width High	tGH	1.5	–	1.5	–	1.5	–	Clk.Per.
46	$\overline{BGACK}$ Width Low (52-Pin Version Only)	tGAL	1.5	–	1.5	–	1.5	–	Clk.Per.
47	Asynchronous Input Setup Time	tASI	10	–	10	–	10	–	ns
57	$\overline{BGACK}$ High to Control Bus Driven (52-Pin Version Only)	tGABD	1.5	–	1.5	–	1.5	–	Clk.Per.
58[1]	BG High to Control Bus Driven	tGHBD	1.5	–	1.5	–	1.5	–	Clk.Per.

NOTES :

1. For processor will negate $\overline{BG}$ and begin driving the bus again if external arbitration logic negates $\overline{BR}$ before asserting $\overline{BGACK}$.
2. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{BG}$ may be reasserted.

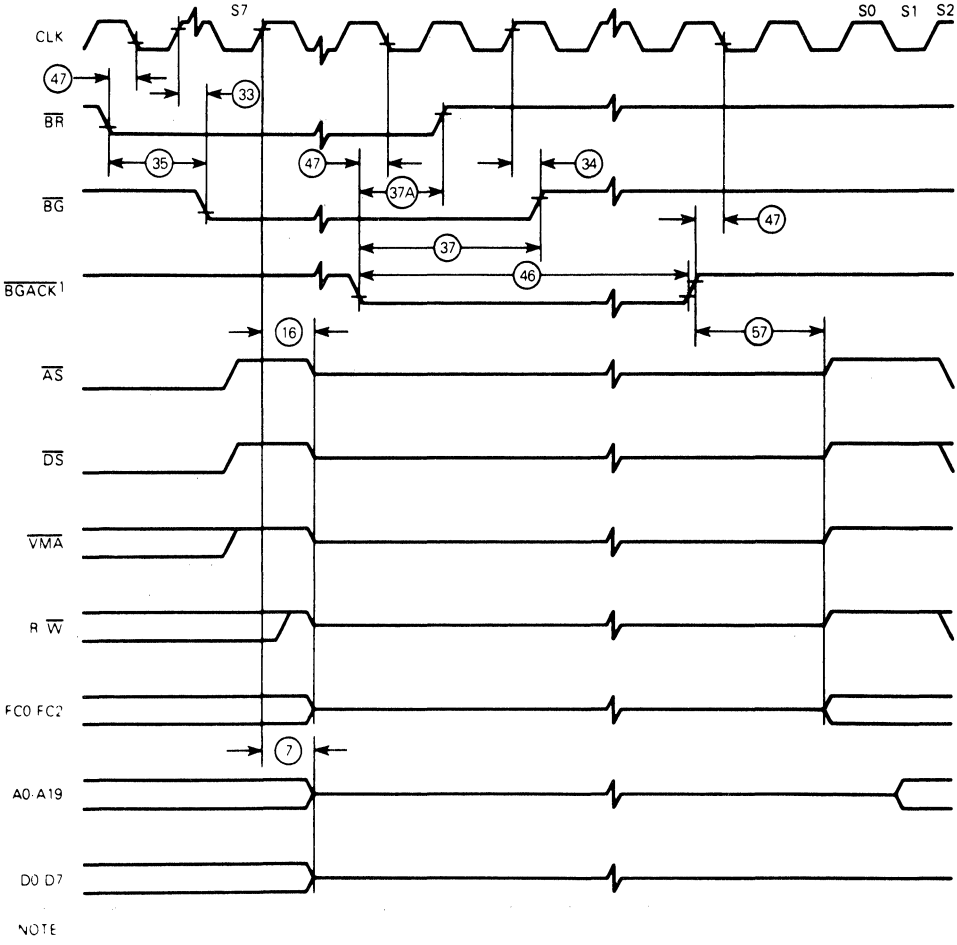
These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.



NOTE:
1. 52-Pin Version of TS68008 Only.

Figure 8-10. Bus Arbitration Timing — Idle Bus Case

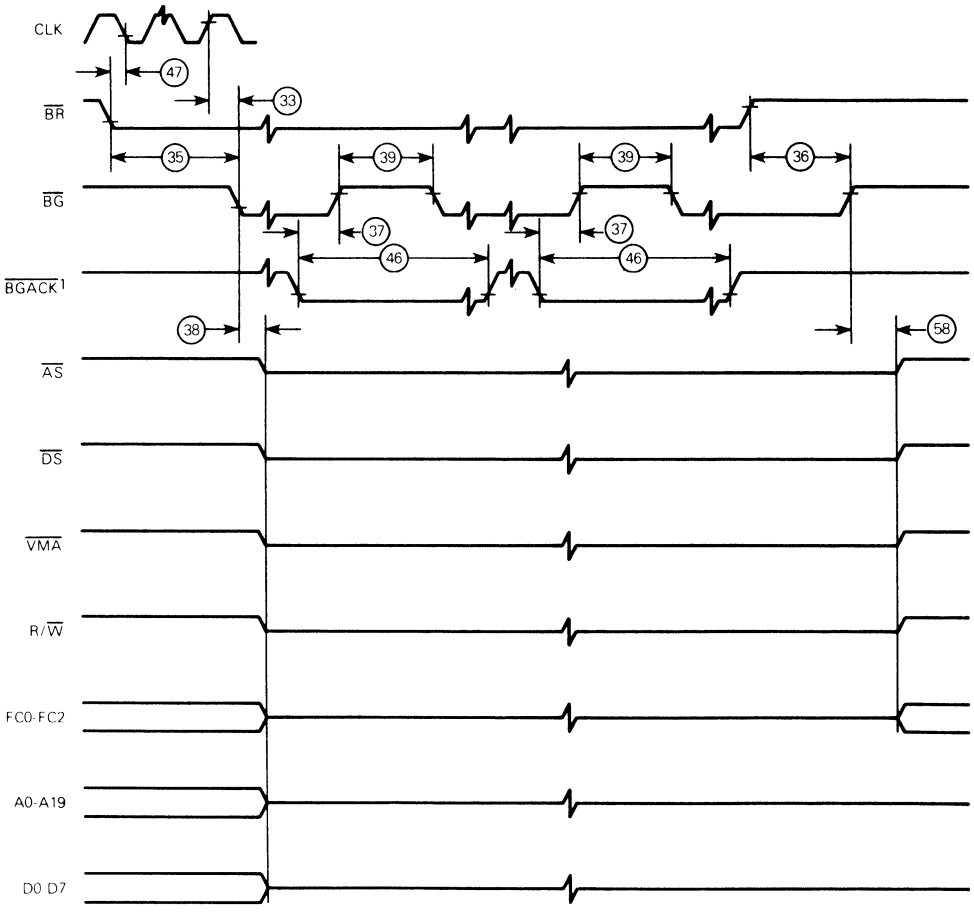
These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.



NOTE
1. 52-Pin Version of TS68008 Only.

Figure 8-11. Bus Arbitration Timing — Active Bus Case

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.



NOTE
1. 52-Pin Version of TS68008 Only.

Figure 8-12. Bus Arbitration Timing — Multiple Bus Requests (52-Pin Version Only)

SECTION 9
ORDERING INFORMATION

This section contains detailed information to be used as a guide when ordering the TS 680008

9.1. STANDARD VERSIONS

Package Type	Frequency (MHz)	Temperature Range	Part Number
Ceramic DIL	8.0	0°C to + 70°C	TS 68008 CC8
C Suffix	8.0	−40°C to + 85°C	TS 68008 VC8
	10.0	0°C to + 70°C	TS 68008 CC10
	10.0	−40°C to + 85°C	TS 68008 VC10
	12.5	−40°C to + 85°C	TS 68008 VC12
Plastic DIL	8.0	0°C to + 70°C	TS 68008 CP8
P. Suffix	8.0	−40°C to + 85°C	TS 68008 VP8
	10.0	−0°C to + 70°C	TS 68008 CP10
	10.0	−40°C to + 85°C	TS 68008 VP10
	12.5	−0°C to + 70°C	TS 68008 CP12
PLCC	8.0	−0°C to + 70°C	TS 68008 CFN8
FN Suffix			

9.2. HI-REL VERSIONS

In order to fit more closely to customer specific requirements, THOMSON SEMICONDUCTEURS is proposing different screening levels for its HI-REL ranges.

G/B screening : Available only from THOMSON SEMICONDUCTEURS, this quality level, very close to the MIL-STD-883, is a cost effective alternative for customers who want to buy HI-REL devices (low guaranteed AQL). The G/B level is in full accordance with the NFC 96883 class G.

B/B screening : Full accordance with the MIL-STD-883 Rev.C, class B (US), the CECC 90.000, class B (European) and with the NFC 96883 class B (French).

Details on screening procedures for these levels of selection are available on request (please contact ou sales representatives).

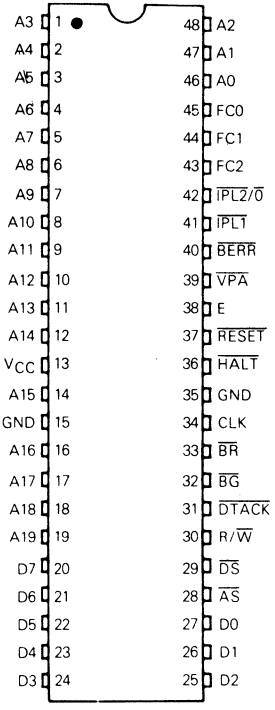
Package Type	Frequency (MHz)	Temperature Range	Part Number
Ceramic DIL	8.0	−40°C to + 85°C	TS 68008 VCG/B8
C Suffix	10.0	−40°C to + 85°C	TS 68008 VCG/B10
	12.5	−40°C to + 85°C	TS 68008 VCG/B12

SECTION 10
MECHANICAL DATA

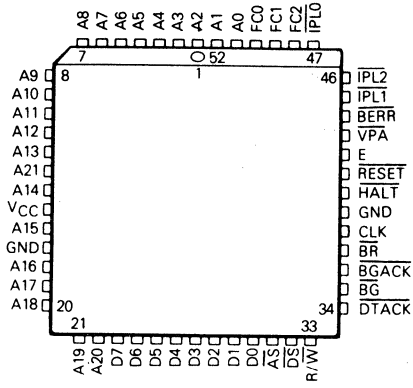
This section contains the pin assignments and package dimensions for the TS68008.

10.1 PIN ASSIGNMENTS

48-Pin Dual-in-Line
(Top View)

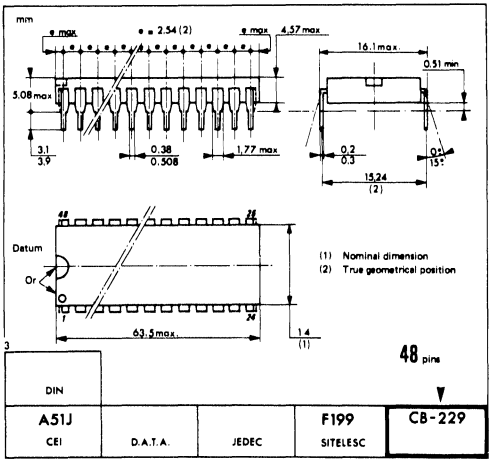


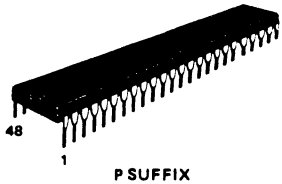
52-Pin Quad Pack (PLCC)
(Top View)



10.2 PACKAGE DIMENSIONS

CB-229





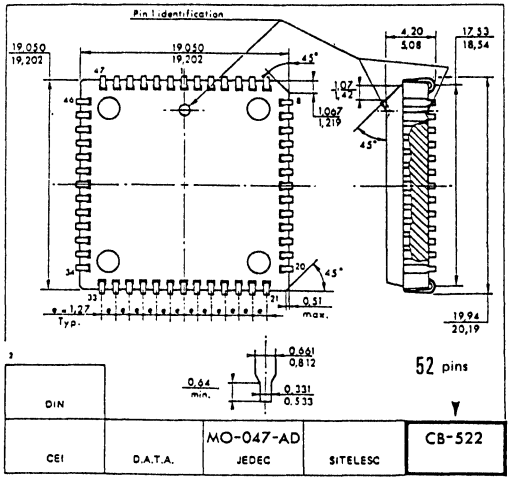
P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE

C SUFFIX
CERAMIC PACKAGE

J SUFFIX
CERDIP PACKAGE

CB-522



FN SUFFIX
PLCC 52

THOMSON COMPONENTS
MOSTEK

MK68200 16-BIT

SINGLE-CHIP MICROCOMPUTERS

MK68201/MK68E201/MK68211/MK68E211/MK68E221

MICROCOMPUTER COMPONENTS

FEATURES

- ☐ 16-bit, high performance, single-chip microcomputer
- ☐ 14 address and data registers
  - — Eight 16-bit or sixteen 8-bit data registers
  - — Six 16-bit address registers
- ☐ Advanced 16-bit instruction set
  - — Bit, byte, and word operands
  - — Nine addressing modes
  - — Byte and word BCD arithmetic
- ☐ High performance (6 MHz instruction clock)
  - — 500 ns register-to-register move or add
  - — 3.5 μs 16 × 16 multiply
  - — 4.0 μs 32/16 divide
- ☐ Available with 0, 4K (2K × 16) of ROM
- ☐ 256 (128 × 16) or 512 (256 × 16) byte RAM
- ☐ Up to 1K byte RAM on MK68E221
- ☐ Three 16-bit timers
  - — Interval modes
  - — Event modes
  - — One-shot modes
  - — Pulse and period measurement modes
  - — Two input and two output pins
- ☐ Serial channel
  - — Double-buffered receive and transmit
  - — Asynchronous to 375 Kbps
  - — Synchronous to 1.5 Mbps
  - — Address wake-up recognition and generation
  - — Internal/external baud rate generation
- ☐ Parallel I/O
  - — Up to 40 pins
  - — Direction programmable by bit
  - — One 16-bit or two 8-bit port(s) with handshaking
- ☐ Interrupt controller
  - — 16 independent vectors
  - — Eight external interrupt sources
  - — One non-maskable interrupt
  - — Individual interrupt masking
- ☐ Optional external bus
  - — 16-bit, multiplexed address/data bus
  - — Automatic bus request/grant arbitration

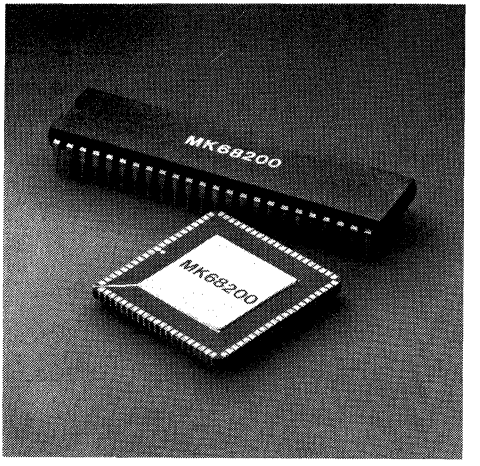


Figure 1. MK68200

  - — Two control bus versions:
    - 68000-compatible bus (UPC)
    - General Purpose Bus (GP)
- ☐ 8 and 12 MHz time base versions produce 4 and 6 MHz instruction clock rates, respectively.
  - — Crystal or external TTL clock
- ☐ Single +5 volt power supply
- ☐ DIP, chip carrier or pin-grid packaging

GENERAL DESCRIPTION

MK68200 designates a series of new, high-performance, 16-bit, single-chip microcomputers from Thomson - Mostek. Implemented in Scaled Poly-5 NMOS, they incorporate an architecture designed for superior performance in computation-intensive control applications. A modern, comprehensive instruction set (which features both high speed execution and code space efficiency) is combined on-chip with extensive, flexible I/O capabilities. On-chip RAM and optional on-chip ROM are provided within a full 64K byte addressing space.

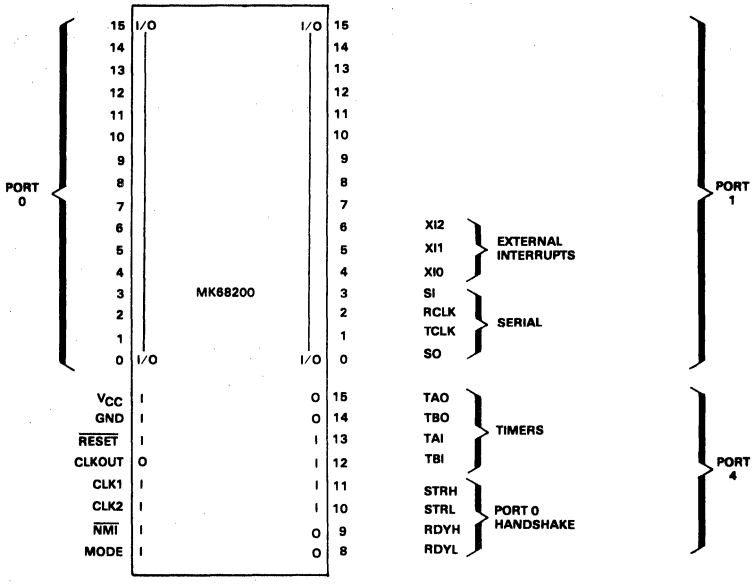


Figure 2. MK68200 Logical Pinout, Single-Chip Mode

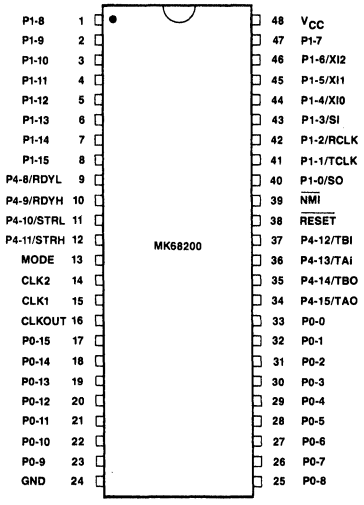


Figure 3. MK68200 Single-Chip Pin Assignment (48-Pin DIP)

The MK68200 is designed to serve the needs of a wide variety of control applications, which require high performance operation with a minimal parts count implementation. Industrial controls, instrumentation, and intelligent computer peripheral controls are all examples of applications served by the MK68200. High speed mathematical ability, rapid I/O addressing and interrupt response, and powerful bit manipulation instructions provide the necessary tools for these applications. In addition to its single-chip microcomputer configuration, both distributed intelligence and parallel multiprocessing system configurations are supported by the MK68200, as illustrated in Figures 13 and 14.

In applications requiring loosely-coupled distributed intelligence, several MK68200's may be interconnected on a common serial network. The on-chip USART supports a wake-up mode in which an additional bit is appended to the data stream to distinguish a serial data word as address or data. The wake-up logic prevents the serial channel from generating interrupts unless certain criteria have been met. The wake-up options available are: Wake-up on any address or data character, wake-up on any address, or wake-up on address match.

Alternately, the MK68200 may be configured as an expandable CPU device which can access external memory and I/O resources. In this operating mode, parallel I/O pins are replaced by multiplexed address/data and control lines. Bus arbitration logic is incorporated on the chip to support a direct interface in parallel shared bus multiprocessor system configurations. Two versions exist which support two types of control signals present on the expanded bus configuration. The General Purpose (GP) bus option allows the MK68200 to operate either as an executive or a peripheral processor. As an executive procesor, the MK68200 can control an external system bus and grant the use of it to requesting devices, such as DMA controllers and/or peripheral processors. As a peripheral control processor, the MK68200 can provide intelligent local control of an I/O device in a computer system and, thereby, relieve the executive processor of these tasks. In this configuration, the MK68200 has the capability of effectively performing DMA transfers between system memory and the I/O device. The on-chip resources of ROM, RAM, and I/O are accessed within the MK68200 without affecting utilization of the shared system bus. Therefore, only external communications compete for bus bandwidth.

The Universal Peripheral Controller (UPC) bus option supports a direct interface to a 68000 executive processor. Thus, the MK68200 can be used as a cost-effective, intelligent peripheral controller in 68000 systems. The UPC version's direct bus interface to the 68000 makes the MK68200 particularly well-suited for performing many intelligent I/O functions in a 68000 system. For example, since the MK68200 includes both a serial channel and an external bus capable of performing DMA transfers, it can be programmed to act as serial protocol controller with DMA capability, as shown in Figure 4.

Table 1 summarizes the specific MK68200 device types that are discussed in this data sheet. A complete guide to the part numbering scheme used throughout this document may be found in the Ordering Information section. All MK68200 devices retain most of the I/O features when they are used in the expanded bus mode; however, 24 pins of parallel I/O are sacrificed when this mode is used. When the expanded bus mode is selected, the MK68201/XX generates UPC (68000-compatible) control signals, while the MK68211/XX generates GP control signals. Also available are 84-pin emulator versions of these devices that do not have on-chip ROM, but instead have additional pins to support a second complete address/data bus to access off-chip ROM, RAM, EPROM, or I/O devices. This bus is referred to as the private bus and is not bonded out on 48-pin versions.

For additional information on the MK68200, refer to the MK68200 Principles of Operation Manual, publication number 4420399.

SINGLE-CHIP DESCRIPTION

Figure 2 illustrates the functions of specific pins for an MK68201 or MK68211, operating in a single-chip mode. When the device is operating in one of the expanded bus modes, the pins on Port 0 become the multiplexed address/data bus, and the upper half of Port 1 becomes the control signals (GP or UPC) for the bus. The following description applies to the pins only when the device is used in the non-expanded or single-chip mode. Descriptions of the pin functions for the expanded bus modes are in the Expanded Bus Operation section of this data sheet.

V_{CC}, GND
(Power, Ground)
Power Supply pins.
(single +5 V)

$\overline{\text{RESET}}$
Input, active low. $\overline{\text{RESET}}$ input overrides ongoing execution (including interrupts) and resets the chip to its initial power-up condition. $\overline{\text{RESET}}$ cannot be masked.

CLKOUT
(Clock Output)
Output. CLKOUT will output the instruction clock rate, which is one-half of the frequency provided on CLK1 and CLK2.

CLK1, CLK2
(Time base Inputs)
Inputs. CLK1 and CLK2 may be connected to a crystal, or CLK1 may be connected to an external TTL-compatible oscillator while CLK2 is left floating. The

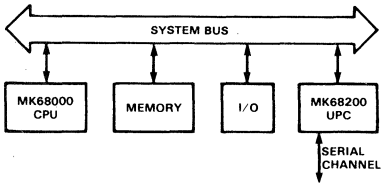


Figure 4. Serial DMA Controller

Table 1. Device Type Summary

Device Type	Expanded Bus Version	ROM (Bytes)	RAM (Bytes)	PKG.
MK68201/04	UPC	0	256	48-pin DIP
MK68201/44	UPC	4K	256	48-pin DIP
MK68E201/04	UPC	0	256	84-pin LCC
MK68211/04	GP	0	256	48-pin DIP, 52-pin PLCC
MK68211/44	GP	4K	256	48-pin DIP, 52-pin PLCC
MK68E211/04	GP	0	256	84-pin LCC
MK68E221/0C	UPC/GP	0	1024	84-pin LCC, PGA

instruction clock rate is one-half of the frequency provided on CLK1 and CLK2.

$\overline{\text{NMI}}$
(Non-Maskable Interrupt)
Input, active low, negative edge triggered. The NMI request line has a higher priority than all of the maskable interrupts. NMI is always enabled regardless of the state of the L1E (Level 1 Interrupt Enable) bit in the Status Register.

MODE
Input. The MODE pin is used to configure the MK68200 on power-up and reset to one of the following states:

Mode Pin
V_{CC} - No expansion (single chip mode)
GND - Partial Expansion
CLKOUT - Full Expansion

P0-0 - P0-15
(Port 0)
Input/Output. Each bit in Port 0 may be individually programmed for general purpose input or output. Port 0 also has several handshaking modes to allow parallel, asynchronous communication with other devices. The high and low bytes may be programmed individually or jointly to be inputs, outputs, or bidirectional.

P1-0 - P1-15
(Port 1)
Input/Output. Each of the 16 bits in Port 1 may be individually programmed for input or output. Additionally, the lowest seven bits of Port 1 may be programmed to serve specific alternate functions, as listed below.

P1-6/XI2
(External Interrupt 2)
Input, rising or falling edge triggered. The programmer may select the rising or falling edge as active for XI2.

P1-5/XI1
(External Interrupt 1)
Input, fixed falling edge triggered. The XI1 interrupt may be used to interrupt the MK68200 on the falling edge of an input pulse.

P1-4/XI0
(External Interrupt 0)
Input, low level triggered. The XI0 interrupt input is level-triggered (unlike XI1, XI2). It may be used to produce an internally vectored interrupt or to cause an external fetch of an interrupt vector number when the MK68200 is used in an expanded mode with the GP bus.

P1-3/SI
(Serial Input)
Input, active high. SI is used to input receive serial data when the receiver is enabled.

P1-2/RCLK
(Receive Clock)
Input/Output, active high. Depending on the mode programmed, RCLK can be used by the serial port as either an input or an output pin. When used as an input pin, RCLK provides the receive clock and/or the transmit clock. When RCLK is not providing the transmit or receive clock, it can be used as an output for Timer C. In this mode, the receive clock is being provided by Timer C.

P1-1/TCLK
(Transmit-Clock)
Input/Output, active high. Depending on the mode programmed, TCLK can be used by the serial port as either an input or an output pin. When used as an input pin, TCLK provides the transmit clock. When TCLK is not providing the transmit clock, it can be used as an output for Timer C. In this mode, the transmit clock is being provided by either Timer C or RCLK.

P1-1/SO
(Serial Output)
Output, active high. SO is used to output transmit serial data when the transmitter is enabled.

P4-8 - P4-15
(Port 4)
Inputs and Outputs. P4-8, P4-9, P4-14, and P4-15 may be used as general purpose outputs, and P4-10, P4-11, P4-12, and P4-13 may be used as general purpose inputs. Interrupts may be generated on the positive transitions on P4-10 and P4-11. Depending on the mode selected, interrupts may be generated on the positive or negative transitions on P4-12, or they may be generated on the positive, negative, or combined transitions on P4-13. Additionally, these bits may be programmed to serve specific alternate functions, as listed below.

P4-15/TAO
(Timer A Output)
Output. TAO may be programmed for special functions in the interval, event, and pulse modes for Timer A. In the interval mode, TAO's state is determined by the Timer A latch (high or low) that is currently active. That is, if the counter is using the high latch for comparison, TAO is high. If the counter is using the low latch for comparison, TAO is low. In the event mode, TAO is initialized to a "1" state and toggles each time the counter matches the Timer A high latch. In the pulse/period modes, TAO is initiated to a "1" state and toggles on positive transitions on TAI.

P4-14/TBO
(Timer B Ouput)
Output. TBO may be programmed for special functions in the interval and one-shot modes for Timer B. In the interval mode, TBO is initialized to a "1" state and toggles each time the counter matches the Timer B latch value. In the one-shot modes, TBO is initialized to a "1" state, and the counter begins counting in response to the occurrence of an active edge on TBI. TBO will not go low until the counter matches the value loaded into the Timer B latch.

P4-13/TAI
(Timer A Input)
Input, positive and/or negative edge triggered. TAI may be programmed for special functions in the event mode or pulse/period modes for Timer A. In the event mode, the counter is incremented on each active transition (positive or negative edge programmable) on TAI. In the pulse/period modes, the counter measures the time during which the signal on TAI remains high and low.

P4-12/TBI
(Timer B Input)
Input, positive or negative edge triggered. TBI may be programmed for special functions for the Timer B one-shot modes. In the one-shot modes, TBI acts as a trigger input.

P4-11/STRH, P4-10/STRL
(Strobe High Byte, Strobe Low Byte)
Input, active high. STRH and STRL are both used for input, output, and bidirectional handshaking on Port 0.

1) Output mode: The positive edge of this strobe is issued by the peripheral to acknowledge the receipt of data made available by the MK68200.

2) Input mode: The strobe is issued by the peripheral to load data from the peripheral into the Port 0 input register. Data is latched into the MK68200 on the negative edge of this signal.

3) Bidirectional mode: When the STRH signal is active, data from the Port 0 output register is gated onto the Port 0 bidirectional data bus.

The negative edge of STRH acknowledges the receipt of the output data. The negative edge of the signal applied to the STRL signal is used to latch input data into Port 0.

P4-9/RDYH, P4-8/RDYL
(Ready High Byte, Ready Low Byte)
Output, active high. RDYH and RHYL are used for input, output, and bidirectional handshaking on Port 0.

1) Output mode: The ready signal goes active to indicate that the Port 0 output register has been loaded, and the peripheral data is stable and ready for transfer to the peripheral device.

2) Input mode: The ready signal is active when the Port 0 input register is empty and is ready to accept data from the peripheral device.

3) Bidirectional mode: The RDYH signal is active when data is available in Port 0 output register for transfer to the peripheral device. In this mode, data is not placed on the Port 0 data bus unless STRH is active. The RDYL signal is active when the Port 0 input register is empty and is ready to accept data from the peripheral device.

PROCESSOR ARCHITECTURE

The MK68200 microcomputer contains an advanced processor architecture, combining the best properties of both 8- and 16-bit processors. A large majority

of instructions operate on either byte or word operands. Figure 5 summarizes the internal architecture of the MK68200.

REGISTERS

The MK68200 register set includes three system registers, six address registers, and eight data registers. The three 16-bit system registers (Figure 6) include a Program Counter, a Status Register, and a Stack Pointer. The six address registers may be used either for 16-bit data or for memory addressing. The eight 16-bit data registers are used for data and may be referenced as sixteen 8-bit registers, providing great flexibility in register allocation.

ADDRESSING

The MK68200 directly addresses a 64K byte memory space, which is organized as 32K 16-bit words. The memory is byte-addressable, but most transfers occur 16 bits at a time, for increased performance over 8-bit microcomputers. All input/output is memory-mapped, and the on-chip I/O is situated in the top 1K bytes of the address space. In the single-chip mode, all resources including ROM, RAM, and I/O, are accessed via an internal or private bus. The memory map, which is accessed by this bus in the single-chip mode, is depicted in Figure 7. Note on-chip RAM always begins at $FBFF and extends downward. ROM always begins at zero and extends upward.

Nine addressing modes provide ease of access to data in the MK68200, as depicted in Table 2. The four register indirect forms utilize the address registers and the Stack Pointer and support many common data structures such as arrays, stacks, queues, and linked lists. I/O Port addressing is a short form addressing mode for the first 16 words of the I/O port space and allows most instructions to access the most often referenced I/O ports in just one word. Many microcomputer applications are I/O intensive and short, fast addressing of I/O has a significant impact on performance.

INSTRUCTION SET

The MK68200 instruction set has been designed with regularity and ease of programming in mind. In addition, instructions have been encoded to minimize code space, a feature which is especially important in single-chip microcomputers. Small code space is related to execution speed, and most instructions execute in either three or six instruction clock periods. (An instruction clock period is equal to 167 ns with a 6 MHz instruction clock). See Table 3.

In addition to operations on bytes and words, the MK68200 has rapid bit manipulation instructions that can operate on registers, memory, and ports. The bit to be affected may be an immediate operand of the instruction, or it may be dynamically specified in a register. Operations available include bit set, clear, test, change, and exchange; and all bit operations perform a bit test as well. Since each instruction is indivisible, this provides the necessary test-and-set function for the implementation of semaphores.

The MOVE group of instructions has the most extensive capabilities. A wide variety of addressing mode combinations is supported including memory-to-memory transfers. A special move multiple is included to save and restore a specified portion of the registers rapidly.

In total, the MK68200 instruction set provides a programming environment, similar to the 68000, which has been optimized for the needs of the single-chip microcomputer marketplace. A summary of the instruction set is provided in Table 4.

Table 2. Addressing Modes

Register
Register Indirect
Register Indirect with Post-increment
Register Indirect with Pre-decrement
Register Indirect with Displacement
Program Counter Relative
Memory Absolute
Immediate
I/O Port

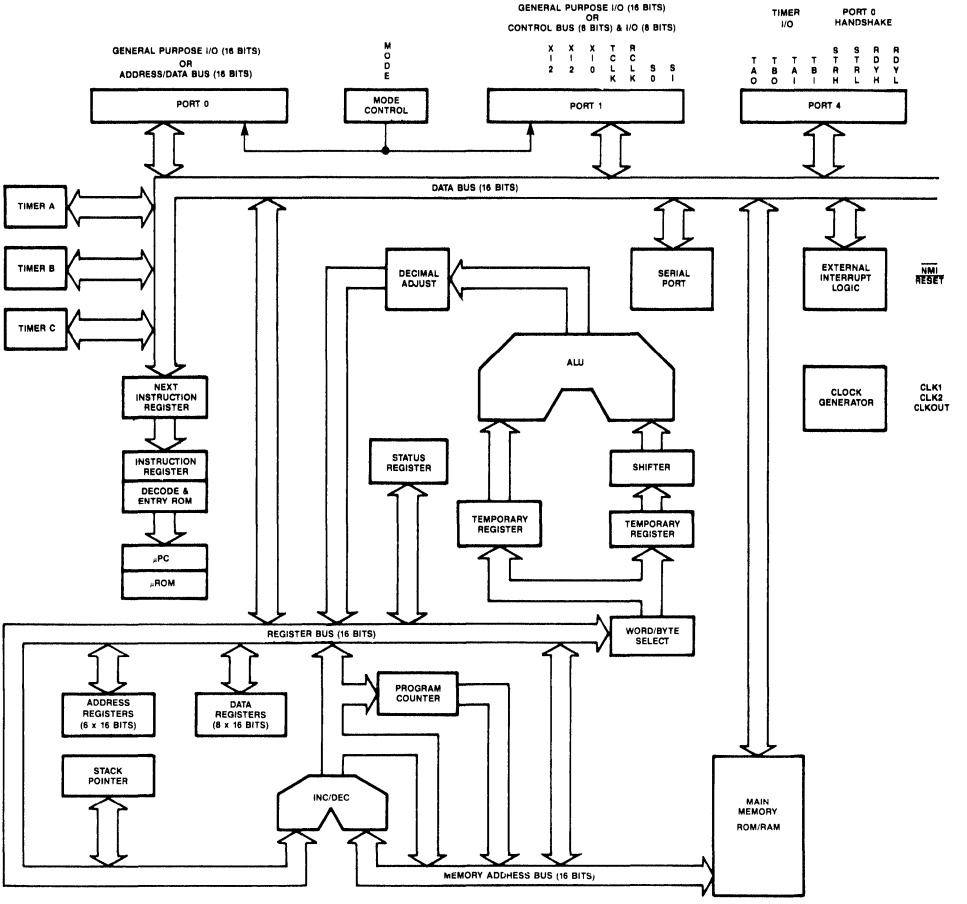
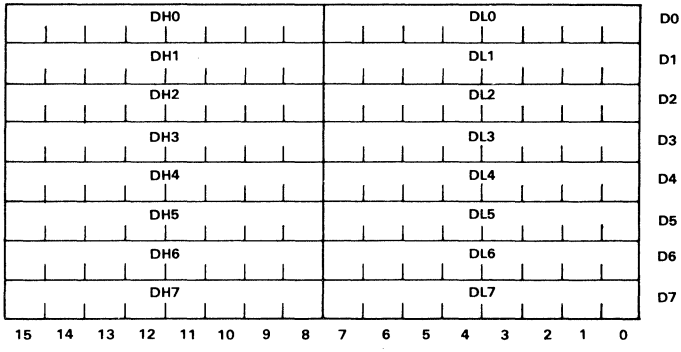
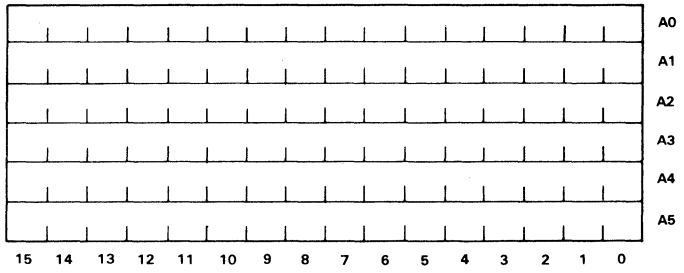


Figure 5. MK68200 Block Diagram

DATA REGISTERS:



ADDRESS REGISTERS:



SYSTEM REGISTERS:

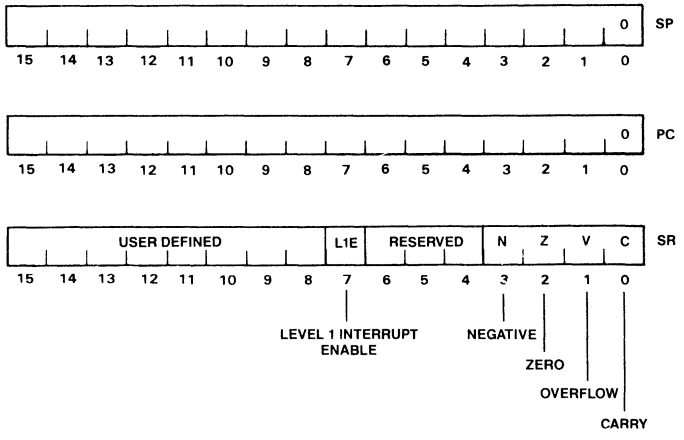


Figure 6. Register Set

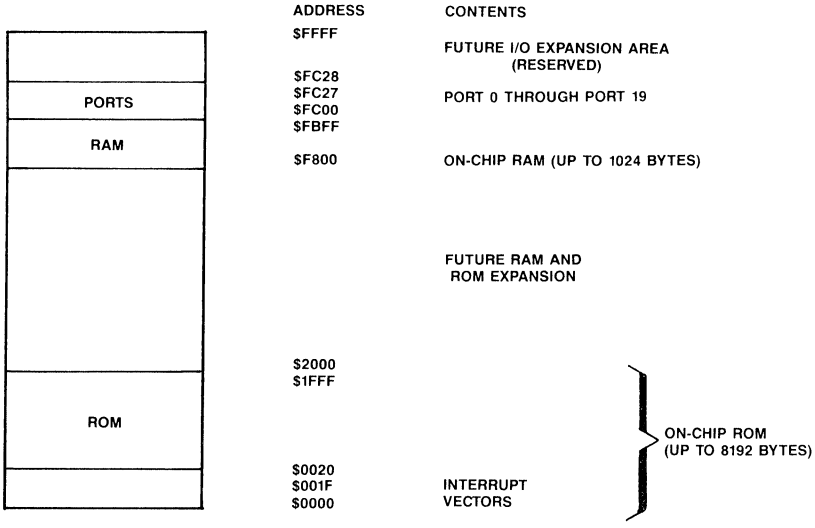


Figure 7. Addressing Space For Single-Chip Configuration

Table 3. Instruction Execution Times

Instruction Type	Clock Periods	Execution Time with 6 MHz Clock (μS)
Move Register-to-register	3	0.5
Add Register-to-register (binary or BCD)	3	0.5
Move Memory-to-register	6	1.0
Add Register-to-memory	9	1.5
Multiply (16 × 16)	21	3.5
Divide (32/16)	23	3.84
Move Multiple (save or restore all registers)	55	9.2

Table 4. Instruction Set Summary

INSTRUCTION	DESCRIPTION	INSTRUCTION	DESCRIPTION
ADD	ADD	HALT	HALT
ADD.B	ADD BYTE	JMPA	JUMP ABSOLUTE
ADDC	ADD WITH CARRY	JMPR	JUMP RELATIVE
ADDC.B	ADD WITH CARRY BYTE	LIBA	LOAD INDEXED BYTE ADDRESS
AND	LOGICAL AND	LIWA	LOAD INDEXED WORD ADDRESSED
AND.B	LOGICAL AND BYTE	LSR	LOGICAL SHIFT RIGHT
ASL	ARITHMETIC SHIFT LEFT	LSR.B	LOGICAL SHIFT RIGHT BYTE
ASL.B	ARITHMETIC SHIFT LEFT BYTE	MOVE	MOVE
ASR	ARITHMETIC SHIFT RIGHT	MOVE.B	MOVE BYTE
ASR.B	ARITHMETIC SHIFT RIGHT BYTE	MOVEM	MOVE MULTIPLE REGISTERS
BCHG	BIT CHANGE	MOVEM.B	MOVE MULTIPLE REGISTERS BYTE
BCLR	BIT CLEAR	MULS	MULTIPLY SIGNED
BEXG	BIT EXCHANGE	MULU	MULTIPLY UNSIGNED
BSET	BIT SET	NEG	NEGATE
BTST	BIT TEST	NEG.B	NEGATE BYTE
CALLA	CALL ABSOLUTE	NEGC	NEGATE WITH CARRY
CALLR	CALL RELATIVE	NEGC.B	NEGATE WITH CARRY BYTE
CLR	CLEAR	NOP	NO OPERATION
CLR.B	CLEAR BYTE	NOT	ONE'S COMPLEMENT
CMP	COMPARE	NOT.B	ONE'S COMPLEMENT BYTE
CMP.B	COMPARE BYTE	OR	LOGICAL OR
DADD	DECIMAL ADD	OR.B	LOGICAL OR BYTE
DADD.B	DECIMAL ADD BYTE	POP	POP
DADDC	DECIMAL ADD WITH CARRY	POPM	POP MULTIPLE REGISTERS
DADDC.B	DECIMAL ADD WITH CARRY BYTE	PUSH	PUSH
DI	DISABLE INTERRUPTS	PUSHM	PUSH MULTIPLE REGISTERS
DIVU	DIVIDE UNSIGNED	RET	RETURN FROM SUBROUTINE
DJNZ	DECREMENT COUNT AND JUMP IF NON-ZERO	RETI	RETURN FROM INTERRUPT
		ROL	ROTATE LEFT
DJNZ.B	DECREMENT COUNT BYTE AND JUMP IF NON-ZERO	ROL.B	ROTATE LEFT BYTE
		ROLC	ROTATE LEFT THROUGH CARRY
DNEG	DECIMAL NEGATE	ROLC.B	ROTATE LEFT THROUGH CARRY BYTE
DNEG.B	DECIMAL NEGATE BYTE		
DNEGC	DECIMAL NEGATE WITH CARRY	ROR	ROTATE BYTE
DNEGC.B	DECIMAL NEGATE WITH CARRY BYTE	ROR.B	ROTATE RIGHT BYTE
		RORC	ROTATE RIGHT THROUGH CARRY
DSUB	DECIMAL SUBTRACT	RORC.B	ROTATE RIGHT THROUGH CARRY BYTE
DSUB.B	DECIMAL SUBRTRACT BYTE		
DSUBC	DECIMAL SUBTRACT WITH CARRY	SUB	SUBTRACT
DSUBC.B	DECIMAL SUBTRACT WITH CARRY BYTE	SUB.B	SUBTRACT BYTE
		SUBC	SUBTRACT WITH CARRY
EI	ENABLE INTERRUPTS	SUBC.B	SUBTRACT WITH CARRY BYTE
EOR	EXCLUSIVE OR	TEST	TEST
EOR.B	EXCLUSIVE OR BYTE	TEST.B	TEST BYTE
EXG	EXCHANGE	TESTN	TEST NOT
EXG.B	EXCHANGE BYTE	TESTN.B	TEST NOT BYTE
EXT	EXTEND SIGN		

INPUT/OUTPUT ARCHITECTURE

The I/O capabilities of the MK68200 are extensive, encompassing timers, a serial channel, parallel I/O, and an interrupt controller. All of these devices are accessible to the programmer as ports within the top 1K bytes of the address space, and the most commonly accessed ports may be accessed with the short port addressing mode. A description of these ports is given in Table 5.

In total, 40 pins of the 48 are used for I/O, and their functions are highly programmable by the user. In particular, many pins can perform multiple functions, and the programmer selects which ones are to be used. For example, TAI may be used as an input for Timer A, an interrupt source, or a general purpose input pin. The interrupt source may be selected simultaneously with either of the other functions.

Table 5. Port Descriptions

PORT	ADDRESS	READ/WRITE	BYTE-ADDRESSABLE	FUNCTION
0	$FC00	READ/WRITE	YES	16 EXTERNAL I/O PINS OR ADDRESS/DATA BUS
1	$FC02	READ/WRITE	YES	16 EXTERNAL I/O PINS (INCLUDING INTERRUPT, SERIAL I/O PINS, AND BUS CONTROL)
2	$FC04	—	—	(RESERVED)
3	$FC06	LOW BYTE: READ/WRITE HIGH BYTE: READ	YES	SERIAL TRANSMIT (LOW BYTE) AND RECEIVE (HIGH BYTE) BUFFER
4	$FC08	INPUTS: READ ONLY OUTPUTS: READ/WRITE	NO	8 EXTERNAL I/O PINS (TIMER CONTROL AND PORT 0 HANDSHAKE CONTROL)
5	$FC0A	—	—	(RESERVED)
6	$FC0C	—	—	(RESERVED)
7	$FC0E	READ/WRITE	NO	INTERRUPT LATCH REGISTER
8	$FC10	READ/WRITE	NO	INTERRUPT MASK REGISTER
9	$FC12	STATUS: READ ONLY CONTROL: READ/WRITE	NO	SERIAL I/O RECEIVE CONTROL AND STATUS
10	$FC14	STATUS: READ ONLY CONTROL: READ/WRITE	NO	SERIAL I/O TRANSMIT CONTROL AND STATUS
11	$FC16	READ GETS COUNTER WRITE GOES TO LATCH	NO	TIMER B LATCH
12	$FC18	READ GETS COUNTER OR LATCH WRITE GOES TO LATCH	NO	TIMER A, LOW LATCH
13	$FC1A	READ GETS COUNTER OR LATCH WRITE GOES TO LATCH	NO	TIMER A, HIGH LATCH
14	$FC1C	READ/WRITE	NO	TIMER CONTROL, INTERRUPT EDGE SELECT
15	$FC1E	READ/WRITE	NO	PORT 0 HANDSHAKE MODE BITS, FAST/STANDARD, BUS LOCK, BUS SEGMENT BITS
16	$FC20	READ/WRITE	NO	PORT 0 DIRECTION CONTROL (DDR0)
17	$FC22	READ/WRITE	NO	PORT 1 DIRECTION CONTROL (DDR1)
18	$FC24	READ/WRITE	NO	SERIAL I/O MODE AND SYNC REGISTER
19	$FC26	READ GETS COUNTER WRITE GOES TO LATCH AND COUNTER	NO	TIMER C LATCH

TIMERS

The MK68200 includes three on-chip timers, each with unique features. They are denoted Timer A, Timer B, and Timer C. All three timers are a full 16 bits in width, and count at the instruction clock rate of the MK68200 processor. Thus, this rate provides a resolution equal to the instruction clock period (tc) of the MK68200. The maximum count interval is equal to tc $*2^{16}$. For a 6 MHz MK68200, a 167 nanosecond clock is provided with a maximum count interval of 10.945 milliseconds. Each timer has the capability to interrupt the processor when it matches a predetermined value stored in an associated latch.

Timer A is capable of operating in interval, event, or two pulse/period modes. There is one 16-bit counter and two 16-bit latches (high and low) associated with Timer A. Once Timer A is initialized in the interval mode, the counter is reset, then increments at the instruction clock rate until the value loaded into the high latch is reached. The counter is then reset, increments until the low latch value is reached, and the cycle is repeated. In the event mode, the counter is incremented for every active edge on TAI (programmable as positive or negative) until the value in the high latch is reached. The counter is then reset, and the cycle repeats. In the pulse/period modes, the times are measured during which the pulse applied stays high and low. The counter is reset on the occurrence of any transition on TAI, and increments at the instruction clock rate until the occurrence of the next transition. The value in the counter at the end of the high level or low level time is loaded into the appropriate latch. Interrupts may be generated each time the counter reaches the high latch or low latch value in the interval mode or when the counter reaches the high latch in the event mode. Also, an interrupt is generated whenever the counter overflows. See the Pin Description section of this data sheet for TAI and TAO functions in the various Timer A modes.

Timer B is capable of operating in interval and one-shot modes. There is one 16-bit counter and one 16-bit latch associated with Timer B. In the interval mode, the counter is initially reset and incremented at the instruction clock rate until the value in the latch is reached. The counter is then reset, and the cycle repeats. In the one-shot modes, the counter begins incrementing in response to an active transition (programmable as positive or negative) on TBI. The counter is reset when the value in the Timer B latch is reached. In the retriggerable one-shot mode, active transitions on TBI always cause the counter to reset and begin incrementing. In the non-retriggerable one-shot mode, active transitions on TBI have no effect until the counter reaches the latch value. Interrupts may be generated each time the counter reaches the latch value. See the Pin Description section of this data sheet for TBI and TBO functions in the various Timer B modes.

Timer C has a 16-bit down counter and latch associated with it and operates only in the interval mode. The output of Timer C toggles each time the counter value rolls over from 0 to the latch value and may be used to internally supply the baud rate clock for the serial port. Also, an interrupt may be generated each time the counter rolls over to the latch value. Timer C may be output on the TCLK pin (P1-3), depending on the mode programmed.

Table 6. Timer Modes

Timer	Modes
A	Interval
A	Event
A	Pulse Width and Period Measurement
B	Interval
B	Retriggerable One-shot
B	Non-retriggerable One-shot
C	Interval
C	Baud Rate Generation

SERIAL CHANNEL

The serial channel on the MK68200 (Figure 8) is a full-duplex USART with double buffering on both transmit and receive. Word length, parity, stop bits, and modes are fully programmable. The asynchronous mode supports bit rates up to 375 Kbps, and the byte synchronous mode operates up to 1.5 Mbps. Either internal or external clocks may be used.

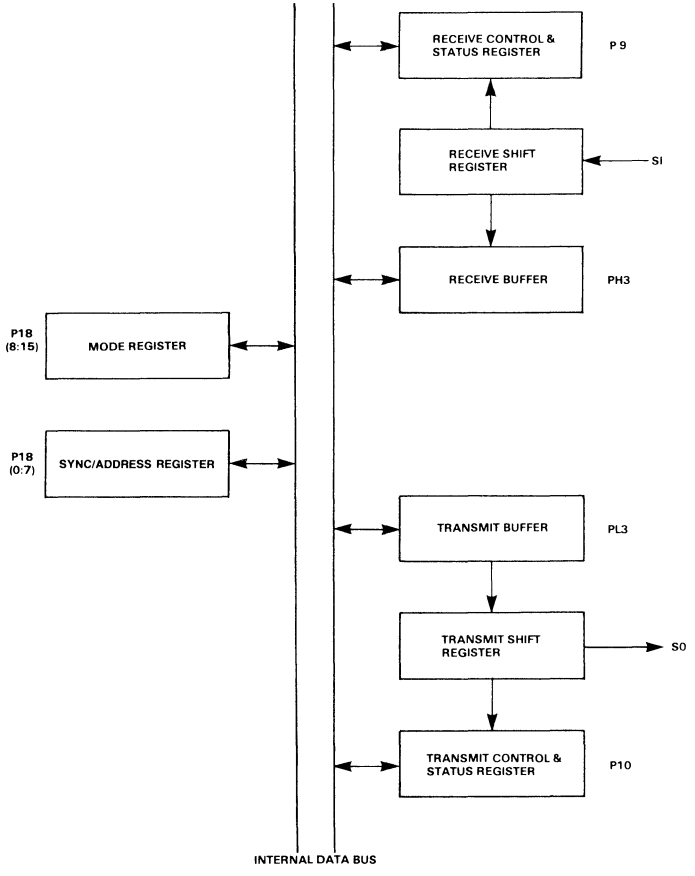
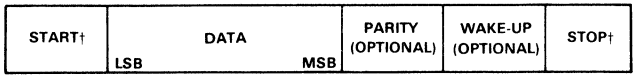


Figure 8. Serial Channel

In addition to the typical USART functions, the serial channel can operate in a special wake-up mode with a wake-up bit appended to each data word, as illustrated in Figure 9. This wake-up bit is used to differentiate normal data words and special address words. The receiver can be programmed to receive only address words or only address words with a specific data value. In this way, the processor can be interrupted only when it receives its particular address and can then change mode to receive the following data words. Wake-up capability is especially useful when several MK68200 microcomputers are interconnected on one serial link.



†USED IN ASYNCHRONOUS MODE ONLY

Figure 9. Serial Frame Format

PARALLEL I/O

Two 16-bit ports, P0 and P1, may be used for parallel I/O. If individual bits are desired, each of the 32 bits may be separately defined as input or output. Bits may be grouped to provide the exact data widths desired. Port 0 has the additional capability of operating under the control of external handshaking signals. Eight- or sixteen-bit sections of P0 may be individually controlled as input, output, or bidirectional I/O. Two pairs of Ready and Strobe signals, which are available as programmable options on Port 4, provide the necessary control.

INTERRUPT CONTROLLER

The MK68200 interrupt controller provides rapid service of up to 15 interrupt sources, each with a unique internal vector. The lowest 16 words of the address space contain the starting addresses of the service routines of each potential interrupt source and reset, as shown in Figure 10.

Interrupt sources and RESET are prioritized in the order shown in Figure 10, with RESET having the highest priority. NMI is the only non-maskable interrupt. All of

VECTOR NUMBER	NAME	MNEMONIC	VECTOR LOCATION	
0	RESET	RESET	0000	
1	NON-MASKABLE INTERRUPT	NMI	0002	LEVEL 2
2	SPARE	SPARE	0004	
3	EXTERNAL INTERRUPT 2	XI2	0006	
4	STROBE LOW	STRL	0008	
5	TIMER A OUTPUT	TAO	000A	
6	TIMER A INPUT	TAI	000C	
7	STROBE HIGH	STRH	000E	
8	RECEIVE SPECIAL CONDITION	RSC	0010	LEVEL 1
9	RECEIVE NORMAL	RN	0012	
A	EXTERNAL INTERRUPT 1	XI1	0014	
B	TIMER B OUTPUT	TBO	0016	
C	TIMER B INPUT	TBI	0018	
D	EXTERNAL INTERRUPT 0	XI0	001A	
E	TRANSMIT	XMT	001C	
F	TIMER C	TC	001E	

Figure 10. Interrupt and Reset Vectors

the other sources share an interrupt enable bit in the processor Status Register. This bit is automatically cleared whenever an interrupt is acknowledged. Also, each of these sources has a corresponding individual mask bit. This feature allows selective masking of particular interrupts, including the ability to choose (with minimal software overhead) any priority scheme desired. In fact, 15 levels of nested priority may be programmed.

EXPANDED BUS OPERATION

When it is necessary to expand beyond the on-chip complement of RAM, ROM, or I/O, or when operation in a parallel multiprocessing system is desired, the MK68200 may be placed in an external bus mode. The MODE pin is used to select the expansion capability on reset. The MODE pin has three states, which select fully expanded external bus, partially expanded external bus, or no expanded bus (single-chip configuration). The MK68200 may also be reconfigured dynamically through software. In an expansion mode, Port 0 becomes the 16-bit multiplexed, address/data bus, and eight bits from Port 1 become control signals which handle data transfer and bus arbitration. Sixteen lines are still available for I/O functions, including eight lines from Port 1 and all eight lines of Port 4.

As shown in Figure 11, two different control bus versions are available: a Universal Peripheral Controller (UPC),

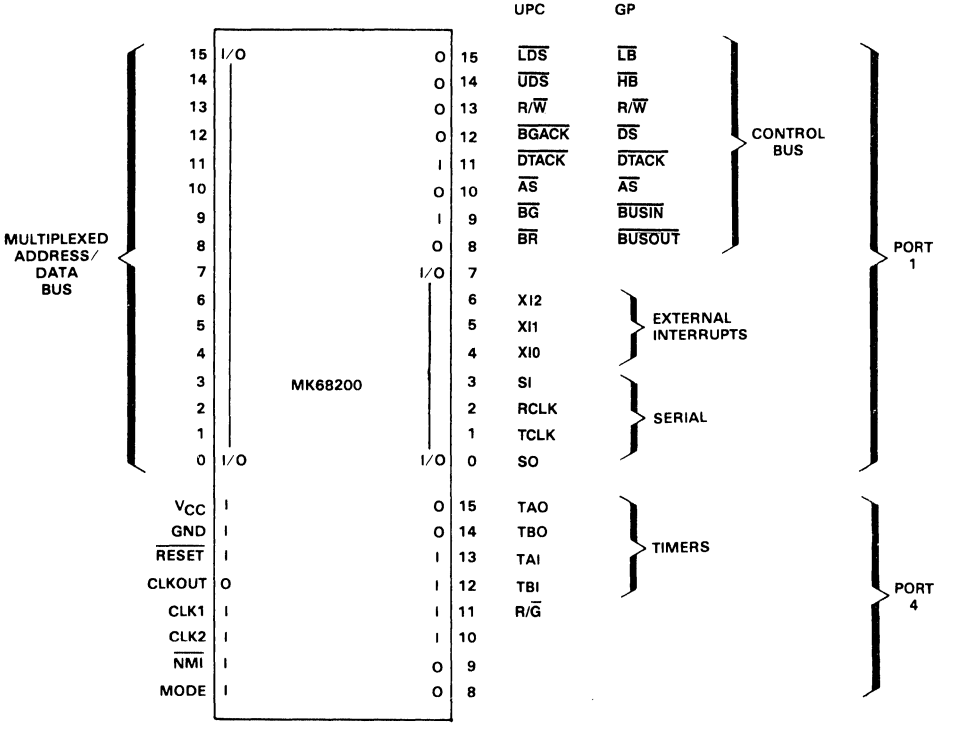


Figure 11. MK68200 Logical Pinout Expanded Bus

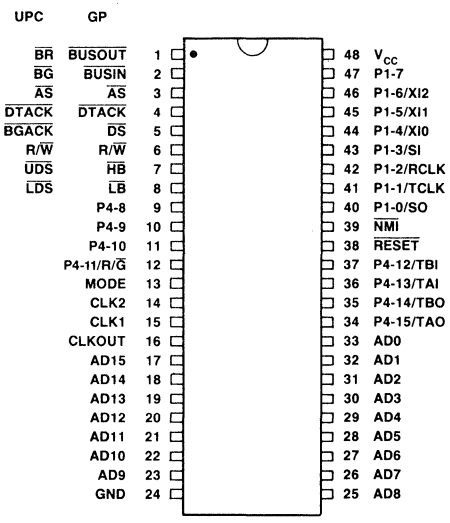


Figure 12. MK68200 Expanded Bus

which generates 68000-compatible signals, and a General Purpose (GP) bus, which can be used to interface to a wide variety of existing microprocessor buses. With the selection of an expanded bus mode, the MK68200 can act either as a general purpose CPU chip (bus grant device) or as an intelligent peripheral I/O controller to a host CPU (bus request device). These two system configurations are illustrated in Figures 13 and 14.

With the GP bus option, the user may configure the MK68200 in either of the two ways shown in Figures 13 and 14. As a host CPU (Figure 13), the MK68200 bus arbitration logic causes the device to act as the system bus grantor. In other words, the MK68200 would normally have control of the system bus and would grant its use to DMA devices or peripheral CPUs. Alternately, the MK68200 may be configured as a peripheral CPU (Figure 14) that must issue a request to the bus grant device before being allowed to use the system bus. The selection of one of these two configurations is accomplished by the P4-11 pin at reset time. During reset, P4-11 serves as the R/$\overline{G}$ input (0=bus grantor, 1 = bus requestor). Following reset and at all times during program execution, P4-11 may be used as a general purpose input pin.

With the GP bus operating in the host CPU configuration, the MK68200 may be used to interface with external memory and I/O devices in a manner that is analogous to any general purpose microprocessor. Additionally, the MK68200 retains its on-chip RAM and I/O resources, with on-chip ROM as an option,

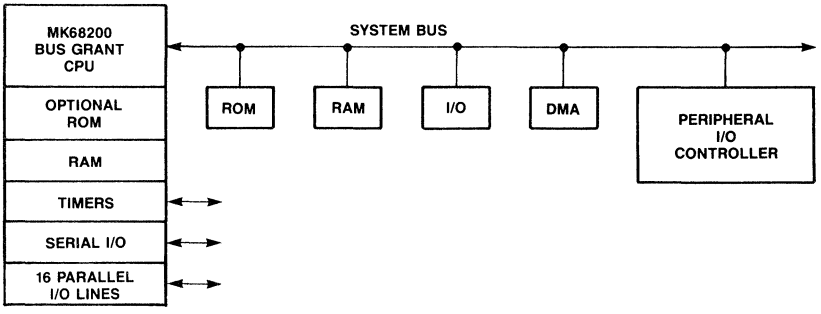


Figure 13. Host CPU Hardware Configuration

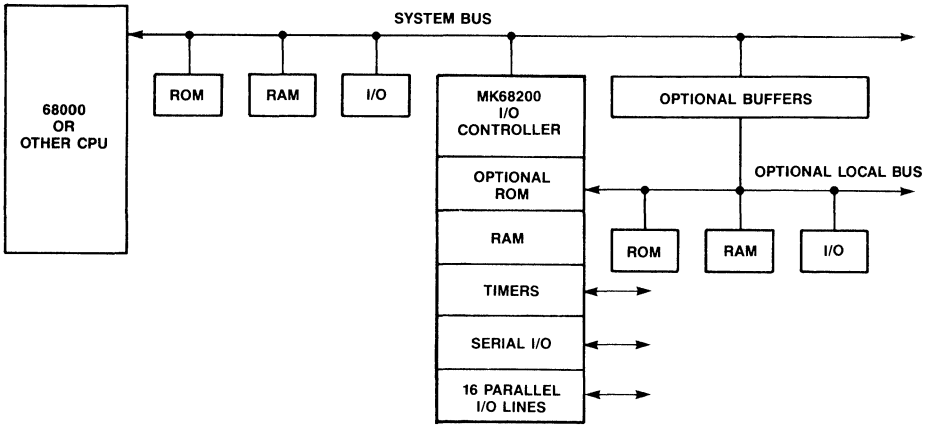


Figure 14. Peripheral I/O Controller Configuration

depending on the expansion configuration selected. $\overline{\text{BUSIN}}$ and $\overline{\text{BUSOUT}}$ are used to perform the bus arbitration handshake function, where $\overline{\text{BUSIN}}$ acts as the bus request input and $\overline{\text{BUSOUT}}$ as the bus grant output.

In the full expansion configuration, any on-chip ROM is disabled, and program memory starting at location $0000 is located off-chip and is addressed via the expanded bus, as shown in Figure 15. In effect, the internal bus from locations $0000-$FAFF is mapped onto the external bus. In the partially expanded configuration (Figure 16), on-chip ROM may be accessed on the internal bus. To gain greater addressability in the partial expansion configuration, a scheme is implemented to allow access of a full 64K-byte address space in four segments on the expanded system bus through the 16K byte "window" on the internal bus. Basically, the most significant two bits of address on the expanded bus are replaced with two user-defined segment bits available to the programmer in an internal I/O control port location.

As a peripheral I/O controller, the MK68200 operates as a bus requestor that gains mastership of the system bus from the bus grant CPU. The GP bus version may be selected to implement this system configuration in cases where an interface to a general purpose CPU is desired. In this case, the $\overline{\text{BUSIN}}$ and $\overline{\text{BUSOUT}}$ lines are again used to perform the bus arbitration handshake function, where $\overline{\text{BUSOUT}}$ now acts as bus request out put, and $\overline{\text{BUSIN}}$ acts as bus grant input. In this configuration, the MK68200 can conceivably act as a complete peripheral I/O control subsystem on a single chip, with 16 lines of I/O and its on-chip ROM, RAM, timers, and serial I/O performing the necessary interface to the I/O device. The UPC bus version provides the peripheral I/O control function with a direct interface to a 68000 bus grant CPU. Note that the UPC bus version can operate only as a bus request device. Once the MK68200 has gained mastership of the system bus via the 68000 bus arbitration handshake lines ($\overline{\text{BR}}$, $\overline{\text{BG}}$, and $\overline{\text{BGACK}}$), it may proceed to perform DMA transfers and communicate with system memory or other I/O devices in the system.

As in the case of the GP bus grant configuration, the portion of the internal (or private) bus address space that is mapped onto the expanded bus when the part is operating as either a GP or a UPC bus request device is determined by the expansion configuration selected. In the partial expansion bus requestor case, the

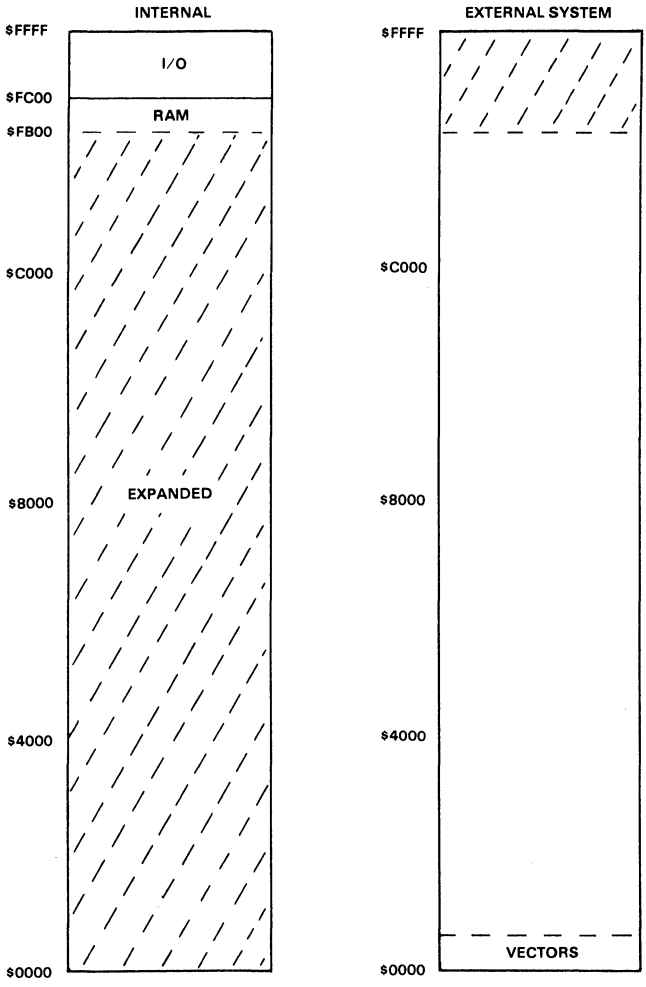


Figure 15. Full Expansion Bus Grantor Memory Map (256 byte RAM version shown)

resulting memory map is identical to that shown for the GP bus grant configuration in Figure 16. During the time the MK68200 is executing its programs from ROM and accessing internal RAM and I/O resources, the expanded bus is held in a tri-state condition. The bus arbitration logic within the MK68200 monitors each memory reference to detect external bus addresses (referenced in segments via the 16K byte DMA window). Whenever such an external reference occurs, the logic automatically holds the processor in a wait state as it proceeds to obtain mastership of the bus. When use of the system bus is obtained, the processor is allowed to continue the reference. This procedure is transparent to the programmer. In case of successive external references, the expanded bus is retained until an internal reference is encountered.

Finally, if the on-chip resources are insufficient to perform the control task in the bus requestor configuration, the internal bus address range (excluding on-chip RAM, I/O) may be mapped onto an external local bus, which is physically the same as the system bus but logically separated with bus buffers. This is the full expansion bus requestor configuration. The memory map for this configuration is shown in Figure 17. The bus arbitration sequence is performed only when the system bus is referenced through the DMA window. In this manner, the I/O subsystem is isolated from the host CPU.

When operating as a bus request device, it is possible to retain the external bus for an indefinite duration by using a bus lock feature. This will help facilitate the transfer of large blocks of data. Thus, the on-chip bus arbitration logic allows (with a minimum of hardware and software overhead) a maximum of concurrent processing in parallel, multiprocessing configurations. The bus lock feature may be used by the MK68200 in a bus grantor mode to keep any peripheral from gaining mastership of the bus.

In any of the GP expanded bus modes, the MK68200 may respond to peripheral devices on the expanded bus which generate an interrupt request on XI0. The MK68200 will obtain the XI0 interrupt vector number from the requesting peripheral on the bus during an interrupt acknowledge cycle. When responding to an interrupt on XI0, the MK68200 will wait for the bus arbitration logic to gain control of the bus and then asserts neither $\overline{HB}$ nor $\overline{LB}$ while asserting $\overline{AS}$ to signify that an interrupt acknowledge cycle is in progress.

Timing diagrams and design parameters for the read, write, and bus arbitration cycles are given in the AC Electrical Specifications section for both the GP and the UPC bus options. Bus timing for the interrupt acknowledge cycle is given for the GP device in the AC Electrical Specifications section. There is a user-programmable speed selection associated with the read and write cycles for both the UPC and GP mask option parts. A bit in an internal I/O port allows the user to select either the standard or the fast read/write cycle on the expanded bus. The standard bus cycle is four clock periods, while the fast bus cycle is three clock periods.

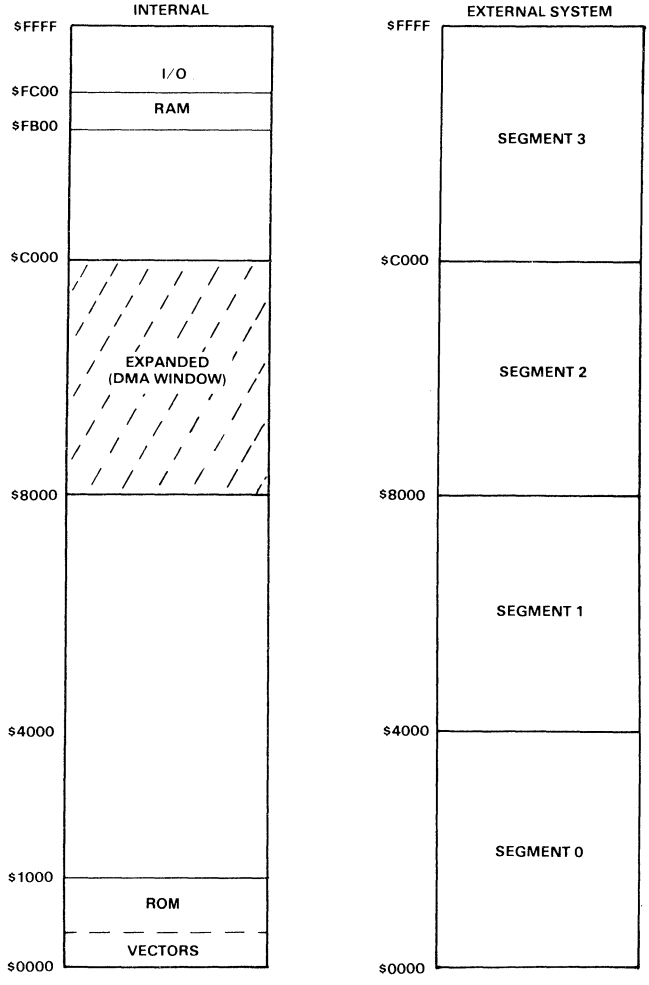


Figure 16. Partial Expansion Memory Map
(256 byte RAM, 4K byte ROM version shown)

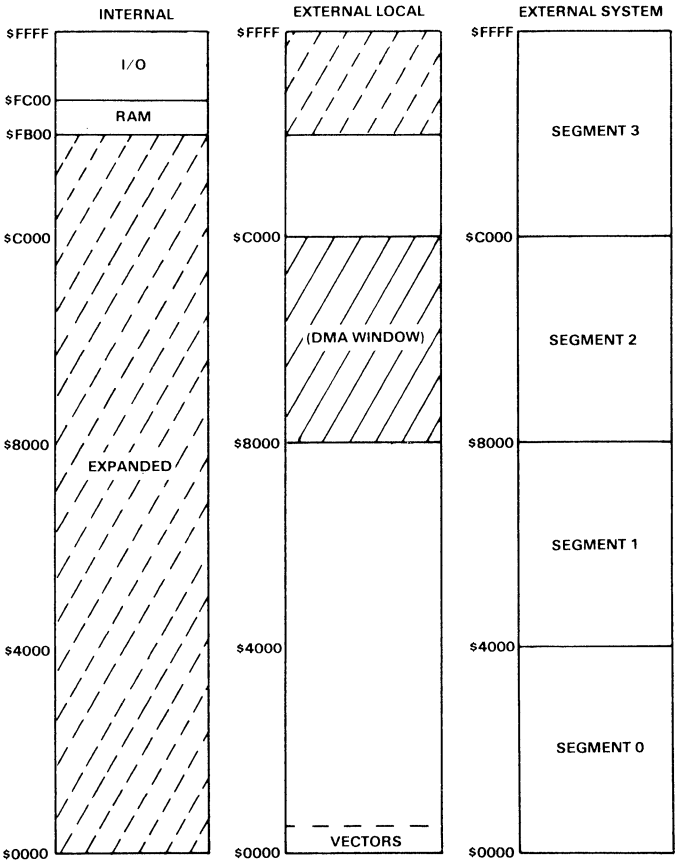


Figure 17. Full Expansion Bus Requestor Memory Map (256 byte RAM version shown)

EXPANDED BUS SIGNALS (Common for GP and UPC Options)

R/$\overline{W}$
(Read/Write)
Output, active high and low. R/$\overline{W}$ determines whether a read or a write is being performed during the current bus cycle. It is stable for the entire bus operation. A high signal denotes a read, and a low signal denotes a write.

$\overline{DTACK}$
(Data Transfer Acknowledge)
Input, active low. When the addressed device has either placed the requested read data on the bus or taken the write data from the bus, $\overline{DTACK}$ should be brought low to signify completion. The data portion of the bus cycle will be extended indefinitely until this signal is asserted. For systems using the GP bus, in which no devices need wait states, $\overline{DTACK}$ may be strapped low.

$\overline{AS}$
(Address Strobe)
Output, active low. $\overline{AS}$ is used to signify that the address is stable on the multiplexed bus. $\overline{AS}$ is high at the beginning of each bus cycle, goes low after the address has stabilized, and returns to the high state near the end of the bus cycle.

UPC BUS SIGNALS

$\overline{UDS}$
(Upper Data Strobe)
Output, active low. $\overline{UDS}$ is used to signify the data portion of the bus cycle for the upper byte of the data bus. For read operations, $\overline{UDS}$ should be used by the external device to gate its most significant byte onto the multiplexed address/data bus. For writes, $\overline{UDS}$ signifies that the upper byte of the bus contains valid data to be written from the processor.

$\overline{LDS}$
(Lower Data Strobe)
Output, active low. $\overline{LDS}$ is used to signify the data portion of the bus cycle for the lower byte of the data bus. For read operations, $\overline{LDS}$ should be used by the external device to gate its least significant byte onto the multiplexed address/data bus. For writes, $\overline{LDS}$ signifies that the lower byte of the bus contains valid data to be written from the processor.

$\overline{BR}$
(Bus Request)
Output, active low, open drain. $\overline{BR}$ goes low when the MK68200 requires external bus master status.

$\overline{BG}$
(Bus Grant)
Input, active low. $\overline{BG}$ notifies that the MK68200 has been granted the external bus master status.

$\overline{BGACK}$
(Bus Grant Acknowledge)
Output, active low, open drain. The MK68200 will assert $\overline{BGACK}$ when it assumes mastership of the system bus.

GP BUS SIGNALS

P4-11 / R/$\overline{G}$
(Request/Grant)
During reset, P4-11 serves as the R/$\overline{G}$ input (0 = bus grantor, 1 = bus requestor). Following reset, and at all times during program execution, P4-11 may be used as a general purpose input pin.

$\overline{DS}$
(Data Strobe)
Output, active low. $\overline{DS}$ is used to signify the data portion of the bus cycle. For read operations, $\overline{DS}$ should be used by the external device to gate its contents onto the multiplexed address/data bus. For writes, $\overline{DS}$ signifies that valid data from the processor is on the bus.

$\overline{HB}$
(High Byte)
Output, active low. $\overline{HB}$ signifies that the upper byte of the data is to be read or written. $\overline{HB}$ remains active for the entire bus cycle.

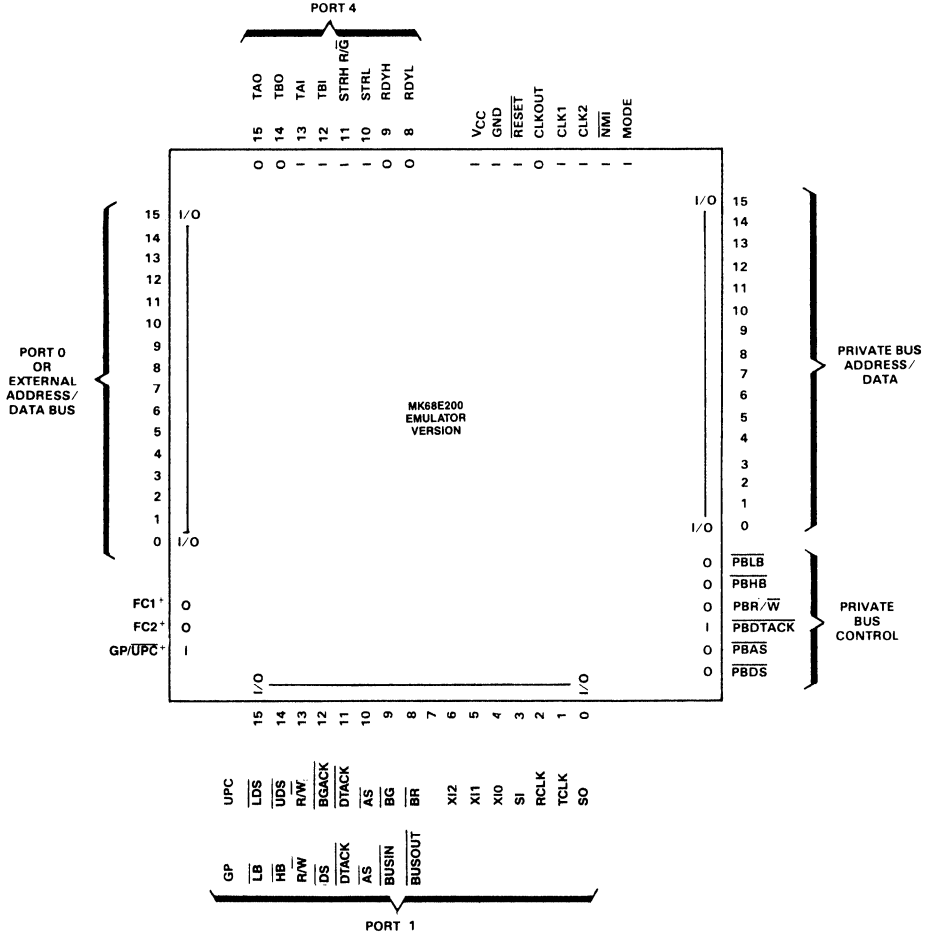
$\overline{LB}$
(Low Byte)
Output, active low. $\overline{LB}$ signifies that the lower byte of the data bus is to be read or written. (Both $\overline{HB}$ and $\overline{LB}$ active imply that an entire word is to be read or written). $\overline{LB}$ remains active for the entire bus cycle.

$\overline{BUSIN}$
(Bus Input)
Input, active low. $\overline{BUSIN}$ provides either bus request or bus grant. When the MK68200 is the bus grant device, its $\overline{BUSIN}$ signal is a bus request input from a requesting device on the bus. When the MK68200 is a bus request device, its $\overline{BUSIN}$ signal is a bus grant from the granting device on the bus.

$\overline{BUSOUT}$
(Bus Output)
Output, active low. $\overline{BUSOUT}$ provides the opposite function of $\overline{BUSIN}$. When $\overline{BUSIN}$ is a bus request signal, $\overline{BUSOUT}$ is the corresponding bus grant, and vice versa.

EMULATOR VERSION

The emulator versions of the MK68200 are available in 84-pin, leadless or leaded chip carrier packages or pin-grid array packages. Figure 18 illustrates the logical pinout of the emulator version. Table 1 summarizes the emulator parts described in this data sheet. The emulator versions have no on-chip ROM, but instead include a second complete bus, referred to as the private bus. The private bus includes a multiplexed address/data bus as well as bus control signals. There are 22 pins associated with the private bus. All 40 I/O port pins that exist on the 48-pin versions are available to the user for configuration either as general purpose or special I/O pins, or as expanded bus pins.



+ AVAILABLE ON MK68E221 (1K RAM VERSION)

Figure 18. MK68E200 Logical Pinout

PRIVATE BUS OPERATION

The address/data lines and control signals that constitute the private bus are functionally equivalent to the internal signals used to access internal resources on the 48-pin versions of the MK68200. Thus, the private bus may be used to interface EPROM memory in emulating mask ROM versions of the MK68200. Alternately, any combination of ROM, RAM, and I/O may reside on the private bus.

The address that is generated on the private bus is identical to that which is internally generated for 48-pin versions. When the part is used in a configuration that supports system bus addressing through the DMA window, any references in this region of the memory map produce an address on the private bus identical to that specified by the programmer. In other words, the segment bits have no effect on the private bus address. Write data appears on the private bus pins for all write operations, regardless of whether the reference is on-chip or off-chip. The MK68200 emulator version reads data from the private bus unless data is read from on-chip RAM, I/O, or the external bus formed by the Port 0 and Port 1 I/O pins.

The I/O port range of the memory map ($FC00-$FFFF) is actually subdivided into space which is exclusively reserved for on-chip I/O ($FC00-$FDFF) and space which is exclusively reserved for in-circuit-emulator, or AIM, use ($FE00-$FFFF). The user should ensure that no external devices reside in the in-circuit-emulator area.

The private bus interface is the same as that for the GP expanded bus. All read/write transfers made exclusively on the private bus are three clock periods, regardless of the state of the Fast/Standard (F/S) bus timing selection bit. The user should ignore all activity on the private bus while accesses are in progress on the expanded bus. Care should also be taken that no external devices reside on the private bus in the memory space intended for expanded bus accesses.

FUNCTION CODE PINS

Function code pins will be available on some versions of the emulator to define the memory cycle currently being executed. They are valid during the time private bus address strobe (PBAS*) is active. The cycle types are interrupt, data fetch, branch, and program fetch. The branch cycle is defined as the first program fetch after a branch occurs. A branch can occur as a result of a jump or call instruction, or an interrupt. For internal interrupts, the interrupt cycles are defined as the two writes to the stack and the read of the vector location which occur during the interrupt acknowledge routine. For external interrupts, the interrupt cycles are defined as the 3 cycles above plus the read of the vector number. The interrupt cycle is a special case of the data fetch cycle. The function code pins are defined below.

TYPE OF CYCLE	FC1	FC2
Interrupt	0	0
Data Fetch	0	1
Branch	1	0
Program Fetch	1	1

CRYSTAL SELECTION

The wide frequency range of crystals that can be chosen for the MK68200 offers the user a large degree of flexibility. To aid in the selection of a suitable crystal, the suggestions shown in Figure 20 should be considered by the user. The MK68200 offers an output pin that will provide a system clock signal at one-half of the crystal frequency.

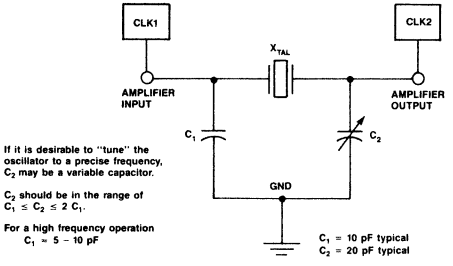


Figure 19. Crystal Connection

FREQUENCY RANGE	SPECIFICATION
1 MHz through 12.0 MHz	PARALLEL RESONANCE FUNDAMENTAL MODE C_L = 20 pF to 40 pF AT CUT

Figure 20. Summary of Crystal Specifications

ASSEMBLER DIRECTIVES

Directive	Function	Assembler Syntax		
DC	Define constant	[label:]	DC[.size]*[1]	expr {,expr}
DS	Define storage	[label:]	DS[.size]*[1]	expr
DUP	Duplicate constant block	[label:]	DUP[.size]*[1]	length, value
END	Program end		END	[start address]
EQU	Equate symbol value	label:	EQU	expr
FAIL	Programmer generated error		FAIL	expr
FORMAT	Format the source listing		FORMAT	
IDNT	Generate module ID	module_name:	IDNT	version, revision
LIST	Enable the assembly listing		LIST	
LLEN	Specify line length		LLEN	length
NOFORMAT	Do not format listing		NOFORMAT	
NOLIST	Disable assembly listing		NOLIST	
NOOBJ	Disable object code generation		NOOBJ	
NOPAGE	Suppress paging		NOPAGE	
OFFSET	Define Offsets		OFFSET	expr
OPT	Assembler output options		OPT	option[2]{, option}
ORG	Define absolute origin		ORG	expr
PAGE	Eject a page in the listing		PAGE	
REG	Define register list	reg_list_name:	REG[.size]	register list
SECTION	Define relocatable program section	[section_name:]	SECTION	number
SET	Set symbol value	label:	SET	expr
SPC	Space between source lines		SPC	number
TTL	Specify heading title string		TTL	title string
XDEF	External symbol definition		XDEF	symbol{, symbol}
XREF	External symbol reference		XREF	[sect no:] symbol - {,[sect no]: symbol}

NOTES:

1. .size = .B or .W (byte or word size)
2. Options for the OPT directive include:
   - CEX — Print DC expansions
   - NOCEX — Do not print DC expansions (default)
   - CL — Print conditional assembly directives (default)
   - NOCL — Do not print conditional assembly directives
   - CRE — Print cross-reference table
   - IMM.L — Forces immediate operands for arithmetic instructions ADD, SUB, DADD, and DSUB to use the long instruction form
   - IMM.S — Allows the assembler to select automatically the short form of the arithmetic instructions for small immediate values (0-15) (default)
   - MC — Print macro calls (default)
   - NOMC — Do not print macro calls
   - MD — Print macro definitions (default)
   - NOMD — Do not print macro definitions
   - MEX — Print macro expansions
   - NOMEX — Do not print macro expansions (default)
   - O — Create object module (default)
   - NOO — Do not create object module
   - STR — Print code generated by structured statements
   - NOSTR — Do not print code generated by structured statements (default)

GENERAL SYMBOL DEFINITIONS

SYMBOL	GENERAL SYMBOL DEFINITIONS
Rn	General Purpose Registers - D0-D7, A0-A5, SP, SR, DH0-DH7, DL0-DL7.
RPn	Register Pairs - D0-D1, D2-D3, D4-D5, D6-D7, A0-A1, A2-A3, A4-A5.
An	Address Registers - A0-A5, SP.
Pn	Ports - P0-P15, PH0-PH3, PL0-PL3.
CC	Condition Code - See Table.
d16	16-Bit Address Displacement Field In Words.
d13	13-Bit Address Displacement Field In Bytes.
d9	9-Bit Address Displacement Field In Bytes.
d8	8-Bit Address Displacement Field In Bytes.
#nx	Immediate Data Field - x Number of Bits.
[s]	Size Bit - '1' = Word, '0' = Byte.
REGn	4-Bit Register Field - See Table.
[PORTn]	4-Bit Port Field - See Table.
[An]	3-Bit Address Register Field - See Table.
[PRTn]	3-Bit Port Field - See Table.
[RGn]	3-Bit Register Pair Field - See Table.
M	Register Mask Field - See Table.
[COND]	Condition Code Field - See Table.
[c3]	3-Bit Class Field - See Table.
[c2]	2-Bit Class Field - See Table
[c1]	1-Bit Class Field—See Table
a	Address Field - 16 Bits.
#	Immediate Data Field.
n	3-Bit Shift Field - $2 \leq n \leq 7$.
b#	4-Bit Bit Select Field.
d	Displacement Field.
.B	Byte Attribute.
.W	Word Attribute.
.L	Long Attribute.
.S	Short Attribute.
[ ]	Optional Field.

FIELD DEFINITIONS

REGn 4-Bit Register Map					
Register		Bit Field	Register		Bit Field
D0	DH0	0 0 0 0	A0	DL0	1 0 0 0
D1	DH1	0 0 0 1	A1	DL1	1 0 0 1
D2	DH2	0 0 1 0	A2	DL2	1 0 1 0
D3	DH3	0 0 1 1	A3	DL3	1 0 1 1
D4	DH4	0 1 0 0	A4	DL4	1 1 0 0
D5	DH5	0 1 0 1	A5	DL5	1 1 0 1
D6	DH6	0 1 1 0	SP	DL6	1 1 1 0
D7	DH7	0 1 1 1	SR	DL7	1 1 1 1

PORTn 4-Bit Port Map				
Port		Bit Field	Port	Bit Field
P0	PH0	0 0 0 0	P8	1 0 0 0
P1	PL0	0 0 0 1	P9	1 0 0 1
P2	PH1	0 0 1 0	P10	1 0 1 0
P3	PL1	0 0 1 1	P11	1 0 1 1
P4	PH2	0 1 0 0	P12	1 1 0 0
P5	PL2	0 1 0 1	P13	1 1 0 1
P6	PH3	0 1 1 0	P14	1 1 1 0
P7	PL3	0 1 1 1	P15	1 1 1 1

An 3-Bit Addr Reg Map	
Register	Bit Field
A0	0 0 0
A1	0 0 1
A2	0 1 0
A3	0 1 1
A4	1 0 0
A5	1 0 1
SP	1 1 0

PTRn 3-Bit Port Map	
Port	Bit Field
PH0	0 0 0
PL0	0 0 1
PH1	0 1 0
PL1	0 1 1
PH2	1 0 0
PL2	1 0 1
PH3	1 1 0
PL3	1 1 1

RGn 3-Bit Reg Pair Map	
Register	Bit Field
D0-D1	0 0 0
D2-D3	0 0 1
D4-D5	0 1 0
D6-D7	0 1 1
A0-A1	1 0 0
A2-A3	1 0 1
A4-A5	1 1 0

M—REGISTER MASK MAP FOR MOVEM, PUSHM, AND POPM																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Inc Word	SR	SP	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
Dec Word	D0	D1	D2	D3	D4	D5	D6	D7	A0	A1	A2	A3	A4	A5	SP	SR
Inc Byte	DH7	DL7	DH6	DL6	DH5	DL5	DH4	DL4	DH3	DL3	DH2	DL2	DH1	DL1	DH0	DL0
Dec Byte	DL0	DH0	DL1	DH1	DL2	DH2	DL3	DH3	DL4	DH4	DL5	DH5	DL6	DH6	DL7	DH7

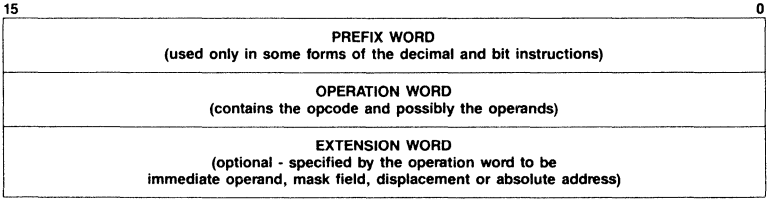
COND CONDITION CODE TABLE			
Condition Code	**Bit Field**	**Description**	**Test**
Z EQ	0 0 0 0	Zero Equal	Z
MI	0 0 0 1	Minus	N
LO[2] CS	0 0 1 0	Lower Carry Set	C
VS	0 0 1 1	Overflow Set	V
GE[2]	0 1 0 0	Greater than or Equal	$\overline{N}$ .EOR. $\overline{V}$
GT[2]	0 1 0 1	Greater than	$\overline{Z}$ .AND. ($\overline{N}$ .EOR. V)
HI[2]	0 1 1 0	Higher	$\overline{C}$ .AND. $\overline{Z}$
F[1]	0 1 1 1	False	Always False
NE NZ	1 0 0 0	Not Equal Not Zero	$\overline{Z}$
PL	1 0 0 1	Plus	$\overline{N}$
HS[2] CC	1 0 1 0	Higher or Same Carry Clear	$\overline{C}$
VC	1 0 1 1	Overflow Clear	$\overline{V}$
LT[2]	1 1 0 0	Less than	N .EOR. V
LE[2]	1 1 0 1	Less than or Equal	Z .OR. (N .EOR. V)
LS[2]	1 1 1 0	Lower or Same	C .OR. Z
T[1]	1 1 1 1	True	Always True

NOTES:
1. The assembler does not recognize the T and F condition codes.
2. LT, LE, GT, and GE are used for unsigned conditions; LO, LS, HI, and HS are for unsigned conditions.

INSTRUCTION CLASS FIELDS									
[c3] - 3-Bit Field			[c2] - 2-Bit Field			[c1] - 1-Bit Field			
Bit Field	Shift Instr	Bit[1] Instr	Bit Field	Arith[2] Instr	Logical Instr	Bit Field	Arith[2] Instr	Test Instr	Neg[2] Instr
0 0 0	ROR	BSET	0 0	ADDC	OR	0	ADD	TESTN	NEGC
0 0 1	ROL	BCHG	0 1	SUBC	EOR	1	SUB	TEST	NEG
0 1 0	RORC	BCLR	1 0	ADD	AND				
0 1 1	ROLC	BTST	1 1	SUB	—				
1 0 0	ASR	—							
1 0 1	ASL	—							
1 1 0	LSR	—							
1 1 1	—	BEXG							

NOTES:
1. The bit fields do not apply to bit instructions using a port operand.
2. These fields also apply to BCD instructions.

INSTRUCTION FORMAT



INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
ADD ADDC SUB SUBC	.B [.W]	Ry,Rx	2 REGx s c2 0 REGy	1	—	3	ADD: Src + Dst → Dst	* * * *
	.B [.W]	(Ay),Rx	2 REGx s c2 1 0 Ay	1	—	6	ADDC: Src + Dst + C → Dst	
	.B [.W]	d16(Ay),Rx	2 REGx s c2 1 1 Ay	2	d	9	SUB: Dst − Src → Dst	
	.B [.W]	Addr,Rx	2 REGx s c2 1 F	2	a	9	SUBC: Dst − Src − C → Dst	
	.B [.W]	#n16,Rx	2 REGx s c2 1 7	2	#	6	Note: For addressing modes #n,Rx and #n,(Ax) with the ADD and SUB instructions, the assembler uses the short version for immediate values ≤4 bits.	
	.B [.W]	Ry,(Ax)	3 0 Ax s c2 0 REGy	1	—	9		
	.B [.W]	(Ax),(Ay)	3 0 Ax s c2 1 0 Ay	1	—	12		
	.B [.W]	#n16,(Ax)	3 0 Ax s c2 1 7	2	#	12		
	.B [.W]	(Ax)+,(Ay)+	3 1 Ax s c2 1 0 Ay	1	—	12		
	.B [.W]	#n16,(Ax)+	3 1 Ax s c2 1 7	2	#	12		
	.B [.W]	−(Ax),−(Ay)	3 1 Ax s c2 1 1 Ay	1	—	12		
	.B [.W]	#n16,−(Ax)	3 1 Ax s c2 1 F	2	#	12		
	.B [.W]	Ry,d16(Ax)	3 1 Ax s c2 0 REGy	2	d	12		
	.B [.W]	Ry,Addr	3 F s c2 0 REGy	2	a	12		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
ADD SUB	.B [.W]	#n4,Rx	B REGx s 1 c1 0 #	1	—	3	ADD: Src + Dst → Dst	* * * *
	.B [.W]	#n4,(Ax)	B 0 Ax s 1 c1 1 #	1	—	9		
	.B [.W]	#n4,d16(Ax)	B 1 Ax s 1 c1 1 #	2	d	12	SUB: Dst − Src → Dst	
	.B [.W]	#n4,Addr	B F s 1 c1 1 #	2	a	12		
AND EOR OR	.B [.W]	Ry,Rx	6 REGx s c2 0 REGy	1	—	3	AND: Src .AND. Dst → Dst	* * 0 0
	.B [.W]	Py,Rx	5 REGx s c2 0 PORTy	1	—	6		
	.B [.W]	(Ay),Rx	6 REGx s c2 1 0 Ay	1	—	6	EOR: Src .EOR. Dst → Dst	
	.B [.W]	d16(Ay),Rx	6 REGx s c2 1 1 Ay	2	d	9		
	.B [.W]	Addr,Rx	6 REGx s c2 1 F	2	a	9	OR: Src .OR. Dst → Dst	
	.B [.W]	#n16,Rx	6 REGx s c2 1 7	2	#	6		
	.B [.W]	Ry,Px	4 PORTx s c2 0 REGy	1	—	9		
	[.W]	#n16,Px	5 PORTx 1 c2 1 9	2	#	12		
	.B [.W]	Ry,(Ax)	7 0 Ax s c2 0 REGy	1	—	9		
Continued on next page	.B [.W]	(Ax),(Ay)	7 0 Ax s c2 1 0 Ay	1	—	12		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
AND EOR OR (cont.)	B [.W]	#n16,(Ax)	7 | 0 | Ax | s | c2 | 1 | 7	2	#	12	AND: Src .AND. Dst → Dst	* * 0 0
	B [.W]	(Ax)+,(Ay)+	7 | 1 | Ax | s | c2 | 1 | 0 | Ay	1	—	12		
	B [.W]	#n16,(Ax)+	7 | 1 | Ax | s | c2 | 1 | 7	2	#	12	EOR: Src .EOR. Dst → Dst	
	B [.W]	(Ax), (Ay)	7 | 1 | Ax | s | c2 | 1 | 1 | Ay	1	—	12		
	B [.W]	#n16,-(Ax)	7 | 1 | Ax | s | c2 | 1 | F	2	#	12	OR: Src .OR. Dst → Dst	
	B [.W]	Ry,d16(Ax)	7 | 1 | Ax | s | c2 | 0 | REGy	2	d	12		
	B [.W]	Ry,Addr	7 | F | s | c2 | 0 | REGy	2	a	12		
ASL ASR LSR ROL ROLC ROR RORC	B [.W]	[#1],Rx	B | REGx | s | 0 | c3 | 0 0 1	1	—	3	Dst → Dst SHIFT: C ← ASL ← 0 0 → LSR → C → ASR → C ROTATE: C ← ROL C ← ROLC → ROR → C → RORC → C *NOTE: 2 ≤ n ≤ 7	* * * * NOTE: The ASR, LSR, ROR, and RORC instructions will clear the V status bit.
	B [.W]	#n3,Rx	B | REGx | s | 0 | c3 | n	1	—	4 + n*		
	B [.W]	(Ax)	B | 0 | Ax | s | 0 | c3 | 0 0 0	1	—	9		
	B [.W]	d16(Ax)	B | 1 | Ax | s | 0 | c3 | 0 0 0	2	d	12		
	B [.W]	Addr	B | F | s | 0 | c3 | 0 0 0	2	a	12		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
BCHG BCLR BEXG BSET BTST	—	#n4,Rx	4 | REGx | c3 | 1 | b#	1	—	3	BCHG: Dst(b#) → C 1 — Dst(b#) → Dst(b#)	* * 0 * If Dst(b#) is in the low byte of the SR,then the other SR bits are unchanged.
	—	#n4,(Ax)	5 | 0 | Ax | c3 | 1 | b#	1	—	9*		
	—	#n4,d16(Ax)	5 | 1 | Ax | c3 | 1 | b#	2	d	12*	BCLR: Dst(b#) → C 0 → Dst(b#)	
	—	#n4,Addr	5 | F | c3 | 1 | b#	2	a	12*		
	—	Ry,Rx	0 | REGy | F | 7	2	—	6	BEXG: Dst(b#) ↔ C	
			4 | REGx | c3 | 1 | 0					
	—	Ry,(Ax)	0 | REGy | F | 7	2	—	12*	BSET: Dst(b#) → C 1 → Dst(b#)	
			5 | 0 | Ax | c3 | 1 | 0					
	—	Ry,d16(Ax)	0 | REGy | F | 7	3	d	15*	BTST: Dst(b#) → C	
			5 | 1 | Ax | c3 | 1 | 0					
	—	Ry,Addr	0 | REGy | F | 7	3	a	15*	*NOTE: The BTST instruction executes in 3 less clock cycles.	
			5 | F | c3 | 1 | 0					

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
BCHG	—	#n4,Px	4 | PORTx | B | b#	1	—	9	Dst(b#) → C 1 — Dst(b#) → Dst(b#)	* * 0 *
	—	Ry,Px	0 | REGy | F | 7	2	—	12		
			4 | PORTx | B | 0					
BCLR	—	#n4,Px	4 | PORTx | D | b#	1	—	9	Dst(b#) → C 0 → Dst(b#)	
	—	Ry,Px	0 | REGy | F | 7	2	—	12		
			4 | PORTx | D | 0					
BEXG	—	#n4,Px	7 | PORTx | E | b#	1	—	9	Dst(b#) → C	
	—	Ry,Px	0 | REGy | F | 7	2	—	12		
			7 | PORTx | E | 0					
BSET	—	#n4,Px	4 | PORTx | 9 | b#	1	—	9	Dst(b#) → C 1 → Dst(b#)	
	—	Ry,Px	0 | REGy | F | 7	2	—	12		
			4 | PORTx | 9 | 0					
BTST Continued on next page	—	#n4,Px	7 | PORTx | 6 | b#	1	—	6	Dst(b#) → C	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
BTST (cont.)	—	Ry,Px	0 | REGy | F | 7	2	—	9	Dst (b#) → C	* * 0 *
			7 | PORTx | 6 | 0					
CALLA	[.L]	(Ax) (unconditional)	5 | 0 Ax | D | F	1	—	9	PC + 2 → –(SP) (Ax) → PC	— — — —
	[.L]	Addr (unconditional)	5 | F | B | F	2	a	9	PC + 4 → –(SP) Addr → PC	
	[.L]	CC, Addr	5 | COND | B | F	2	a	F:6 T:12	If COND is true, PC + 4 → –(SP); Addr → PC	
							NOTE: The initial PC value is the location of the CALLA instruction.	
CALLR	[.S]	d13 (unconditional)	F | d	1	—	10	PC + 2 → –(SP) PC + 4 + 2·(d) → PC	— — — —
	.L	d16 (unconditional)	5 | F | 9 | F	2	d	9	PC + 4 → –(SP) PC + 4 + d → PC	
	.L	CC, d16	5 | COND | 9 | F	2	d	F:6 T:12	If COND is true, PC + 4 → –(SP); PC + 4 + d → PC	
							NOTE: The initial PC value is the location of the CALLR instruction. The displacement, d13, is in signed magnitude representation, and the displacement, d16, is in two's complement representation.	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
CLR	.B [.W]	Rx	1 REGx s 0 1 1 7	1	—	3	0 → Dst	— — — —
	.B [.W]	Px	1 PORTx s 0 1 0 7	1	—	6		
	.B [.W]	(Ax)	1 0 Ax s 0 1 1 F	1	—	6		
	.B [.W]	d16(Ax)	1 1 Ax s 0 1 1 F	2	d	9		
	.B [.W]	Addr	1 F s 0 1 1 F	2	a	9		
CMP	.B [.W]	Ry,Rx	6 REGx s 1 1 0 REGy	1	—	3	Dst — Src	* * * *
	.B [.W]	Py,Rx	5 REGx s 1 1 0 PORTy	1	—	6		
	.B [.W]	(Ay),Rx	6 REGx s 1 1 1 0 Ay	1	—	6		
	.B [.W]	(Ay) + ,Rx	4 REGx s 1 1 0 0 Ay	1	—	6		
	.B [.W]	– (Ay),Rx	4 REGx s 1 1 0 1 Ay	1	—	6		
	.B [.W]	d16(Ay),Rx	6 REGx s 1 1 1 1 Ay	2	d	9		
	.B [.W]	Addr,Rx	6 REGx s 1 1 1 F	2	a	9		
	.B	#n8,Rx	C REGx #	1	—	3		
Continued on next page	[.W]	#n16,Rx	6 REGx F 7	2	#	6		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
CMP (cont.)	.B	#n8,Px	D 0 PRTx #	1	—	6	Dst — Src	* * * *
	[.W]	#n16,Px	4 PORTx E F	2	#	9		
	.B [.W]	(Ax),(Ay)	7 0 Ax s 1 1 1 0 Ay	1	—	9		
	.B [.W]	#n16,(Ax)	7 0 Ax s 1 1 1 7	2	#	9		
	.B [.W]	(Ax) + ,(Ay) +	7 1 Ax s 1 1 1 0 Ay	1	—	9		
	.B [.W]	#n16,(Ax) +	7 1 Ax s 1 1 1 7	2	#	9		
	.B [.W]	– (Ax), – (Ay)	7 1 Ax s 1 1 1 1 Ay	1	—	9		
	.B [.W]	#n16, – (Ax)	7 1 Ax s 1 1 1 F	2	#	9		
DADD DADDC DSUB DSUBC	.B [.W]	Ry,Rx	A REGx s c2 0 REGy	1	—	3	DADD: BCD: [Src + Dst → Dst]	U * U *
	.B [.W]	(Ay),Rx	A REGx s c2 1 0 Ay	1	—	6	DADDC: BCD: [Src + Dst + C → Dst]	
	.B [.W]	d16(Ay),Rx	A REGx s c2 1 1 Ay	2	d	9	DSUB: BCD: [Dst – Src → Dst]	
	.B [.W]	Addr,Rx	A REGx s c2 1 F	2	a	9		
Continued on next page	.B [.W]	#n16,Rx	A REGx s c2 1 7	2	#	6	DSUBC: BCD: [Dst – Src – C → Dst]	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
DADD DADDC DSUB DSUBC (cont.)	.B [.W]	Ry,(Ax)	0 F F F	2	—	12		U * U *
			3 0 Ax s c2 0 REGy				NOTE: For addressing modes #n,Rx and #n,(Ax) with the DADD and DSUB instructions, the assembler uses the short version for immediate values ≤4 bits.	
	.B [.W]	(Ax),(Ay)	0 F F F	2	—	15		
			3 0 Ax s c2 1 0 Ay					
	.B [.W]	#n16,(Ax)	0 F F F	3	#	15		
			3 0 Ax s c2 1 7					
	.B [.W]	(Ax) + ,(Ay) +	0 F F F	2	—	15		
			3 1 Ax s c2 1 0 Ay					
	.B [.W]	#n16,(Ax) +	0 F F F	3	#	15		
			3 1 Ax s c2 1 7					
	.B [.W]	– (Ax), – (Ay)	0 F F F	2	—	15		
			3 1 Ax s c2 1 1 Ay					
	.B [.W]	#n16, – (Ax)	0 F F F	3	#	15		
			3 1 Ax s c2 1 F					

Continued on next page

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
DADD DADDC DSUB DSUBC (cont.)	.B [.W]	Ry,d16(Ax)	0 F F F	3	d	15		U * U *
			3 1 Ax s c2 0 REGy					
	.B [.W]	Ry,Addr	0 F F F	3	a	15		
			3 F s c2 0 REGy					
DADD DSUB	.B [.W]	#n4,Rx	0 F F F	2	—	6	DADD: BCD: [Src + Dst + C → Dst]	U * U *
			B REGx s 1 c1 0 #					
	.B [.W]	#n4,(Ax)	0 F F F	2	—	12	DSUB: BCD: [Dst – Src – C → Dst]	
			B 0 Ax s 1 c1 1 #					
	.B [.W]	#n4,d16(Ax)	0 F F F	3	d	15		
			B 1 Ax s 1 c1 1 #					
	.B [.W]	#n4,Addr	0 F F F	3	a	15		
			B F s 1 c1 1 #					
DI	—	—	4 F 5 7	1	—	3	0 → L1E ; Disable Interrupts	— — — —

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
DIVU	—	Rx,RPy	5 | REGx | D | RGy | 0	1	—	F:6 T:23	RPy(even),RPy(odd)/Rx → RPy(even) REM → RPy(odd) MSW = even; LSW = odd	— — * —
DJNZ	.B [.W]	Rx,d8	9 | REGx | s | d	1	—	F:6 T:9	Rx − 1 → Rx; If Rx ≠ 0 then PC + 4 + [2•(d)] → PC NOTE: The initial PC value is the location of the DJNZ instruction. The displacement value, d, is a magnitude.	— — — —
DNEG DNEGC	.B [.W]	Ry	0 | F | F | F	2	—	6	DNEG: BCD: [0 − Dst → Dst]	U * U *
			3 | 7 | s c1 1 0 | REGy					
	.B [.W]	(Ay)	0 | F | F | F	2	—	12	DNEGC: BCD: [0 − Dst − C → Dst]	
			3 | 7 | s c1 1 1 0 | Ay					
	.B [.W]	d16(Ay)	0 | F | F | F	3	d	15		
			3 | 7 | s c1 1 1 1 | Ay					
	.B [.W]	Addr	0 | F | F | F	3	a	15		
			3 | 7 | s c1 1 1 | F					
DSUB DSUBC		see DADD (page 15)						

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
EI	—	—	4 | F | 1 | 7	1	—	3	1 → L1E Enable Interrupts	— — — —
EOR		see AND (page 9)						
EXG	.B [.W]	Rx,Ry	0 | REGx | s 1 0 0 | REGy	1	—	4	Src → Dst	— — — —
	.B [.W]	Rx,(Ay)	0 | REGx | s 1 0 1 0 | Ay	1	—	9		
	.B [.W]	Rx,(Ay) +	0 | REGx | s 1 1 1 0 | Ay	1	—	9		
	.B [.W]	Rx, - (Ay)	0 | REGx | s 1 1 1 1 | Ay	1	—	9		
	.B [.W]	Rx,d16(Ay)	0 | REGx | s 1 0 1 1 | Ay	2	d	12		
	.B [.W]	Rx,Addr	0 | REGx | s 1 0 1 | F	2	a	12		
EXT	—	Rx	0 | REGx | D | 7	1	—	3	SIGN EXTEND: SignBit(RLx) → RHx	— — — —
HALT	—	—	1 | 7 | 3 | F	1	—	3	PC + 2 → PC Stop	— — — —
JMPA	[.L]	(Ax) (unconditional)	5 | 1 Ax | D | F	1	—	6	(Ax) → PC	— — — —
	[.L]	Addr (unconditional)	5 | F | B | 7	2	a	6	Addr → PC	
	[.L]	CC,Addr	5 | COND | B | 7	2	a	F:6 T:9	If COND is true, Addr → PC	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
JMPR	[.S]	d9 (unconditional)	8 F d	1	—	7	PC + 4 + 2•(d) → PC	— — — —
	[.S]	CC,d9	8 COND d	1	—	F:4 T:7	If COND is true, PC + 4 + 2•(d) → PC	
	.L	d16 (unconditional)	5 F 9 7	2	d	9	PC + 4 + d → PC	
	.L	CC,d16	5 COND 9 7	2	d	F:6 T:9	If COND is true, PC + 4 + d → PC	
							NOTE: The initial PC value is the location of the JMPR instruction. The displacement, d16, is in two's complement representation, and the displacement, d9, is in signed magnitude representation.	
LIBA LIWA	—	#d16(Rx),Az	1 REGx s 1 Az 1 1 1	2	#	6	Rx[-2] + d → Az	— — — —
	—	#d16(Ay,Rx),Az	1 REGx s 1 Az Ay	2	#	6	Ay + (Rx[-2]) + d → Az	
							NOTE: [-2] used for LIWA only. REGx and Az must not refer to same register when using 2nd addressing mode.	
LSR		see ASL (page 10)						
MOVE	B [.W]	Ry,Rx	0 REGx s 0 0 0 REGy	1	—	3	Src → Dst	— — — —
	B [.W]	Py,Rx	0 REGx s 0 1 1 PORTy	1	—	6		
Continued on next page	B [.W]	(Ay),Rx	0 REGx s 0 0 1 0 Ay	1	—	6		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
MOVE (cont.)	B [.W]	(Ay)+,Rx	1 REGx s 0 1 1 0 Ay	1	—	6	Src → Dst	— — — —
	B [.W]	-(Ay),Rx	1 REGx s 0 1 1 1 Ay	1	—	6		
	B [.W]	d16(Ay),Rx	0 REGx s 0 0 1 1 Ay	2	d	9		
	B [.W]	Addr,Rx	0 REGx s 0 0 1 F	2	a	9		
	B	#n8,Rx	E REGx #	1	—	3		
	[.W]	#n16,Rx	0 REGx 9 7	2	#	6		
	B [.W]	Ry,Px	0 PORTx s 0 1 0 REGy	1	—	6		
	B [.W]	(Ay)+,Px	0 PORTx s 1 1 0 0 Ay	1	—	9		
	B	#n8,Px	D 1 PRTx #	1	—	6		
	[.W]	#n16,Px	0 PORTx E 7	2	#	9		
	B [.W]	Ry,(Ax)	1 0 Ax s 0 0 0 REGy	1	—	6		
	B [.W]	(Ax),(Ay)	1 0 Ax s 0 0 1 0 Ay	1	—	9		
	B [.W]	#n16,(Ax)	1 0 Ax s 0 0 1 7	2	#	9		
Continued on next page	B [.W]	Rx,(Ay)+	1 REGx s 0 1 0 0 Ay	1	—	6		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
MOVE (cont.)	B [.W]	(Ax) + ,(Ay) +	1 1 Ax s 0 0 1 0 Ay	1	—	9	Src → Dst	— — — —
	B [.W]	#n16,(Ax) +	1 1 Ax s 0 0 1 7	2	#	9		
	B [.W]	Rx, – (Ay)	1 REGx s 0 1 0 1 Ay	1	—	6		
	B [.W]	Px, – (Ay)	0 PORTx s 1 1 0 1 Ay	1	—	9		
	B [.W]	– (Ax), – (Ay)	1 1 Ax s 0 0 1 1 Ay	1	—	9		
	B [.W]	#n16, – (Ax)	1 1 Ax s 0 0 1 F	2	#	9		
	B [.W]	Ry,d16(Ax)	1 1 Ax s 0 0 0 REGy	2	d	9		
	B [.W]	Ry,Addr	1 F s 0 0 0 REGy	2	a	9		
MOVEM	B [.W]	(Ay) + ,REGLIST	1 7 s 0 0 1 0 Ay	2	M	9 + 3n	(Ay) + → REGLIST	— — — —
	B [.W]	– (Ay),REGLIST	1 7 s 0 0 1 1 Ay	2	M	9 + 3n	– (Ay) → REGLIST	
	B [.W]	REGLIST,(Ay) +	1 F s 0 0 1 0 Ay	2	M	7 + 3n	REGLIST → (Ay) +	
	B [.W]	REGLIST, – (Ay)	1 F s 0 0 1 1 Ay	2	M	7 + 3n	REGLIST → – (Ay)	
							NOTE: A minimum of 2 registers must be specified and may be specified in any order.	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
MULS	—	Rx,RPy	5 REGx B RGy 0	1	—	21	Rx * RPy(even) → RPy(even), RPy(odd) MSW → even; LSW → odd	— — — —
MULU	—	Rx,RPy	5 REGx 9 RGy 0	1	—	21	Rx * RPy(even) → RPy(even), RPy(odd) MSW → even; LSW → odd	— — — —
NEG NEGC	B [.W]	Ry	3 7 s c1 1 0 REGy	1	—	3	2's Complement	* * * *
	B [.W]	(Ay)	3 7 s c1 1 1 0 Ay	1	—	9	NEG: 0 – Dst → Dst	
	B [.W]	d16(Ay)	3 7 s c1 1 1 1 Ay	2	d	12	NEGC: 0 – Dst – C → Dst	
	B [.W]	Addr	3 7 s c1 1 1 F	2	a	12		
NOP	—	—	0 0 0 0	1	—	3	PC + 2 → PC	— — — —
NOT	B [.W]	Ry	3 7 s 0 0 0 REGy	1	—	3	1's Complement	* * 0 0
	B [.W]	(Ay)	3 7 s 0 0 1 0 Ay	1	—	9	NOT(Dst) → Dst	
	B [.W]	d16(Ay)	3 7 s 0 0 1 1 Ay	2	d	12		
	B [.W]	Addr	3 7 s 0 0 1 F	2	a	12		
OR		see AND (page 9)						

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
POP	—	Rx	1 REGx B 6	1	—	6	(SP)+ → Rx	— — — —
	—	Px	0 PORTx E 6	1	—	9	(SP)+ → Px	
	—	(Ay)+	1 E 9 0 Ay	1	—	9	(SP)+ → (Ay)+	
POPM	—	REGLIST	1 7 9 6	2	M	9+3n	(SP)+ → REGLIST NOTE: A minimum of 2 registers must be specified and may be specified in any order.	— — — —
PUSH	—	Rx	1 REGx A E	1	—	6	Rx → -(SP)	— — — —
	—	Px	0 PORTx E E	1	—	9	Px → -(SP)	
	—	-(Ay)	1 1 Ax 9 E	1	—	9	-(Ax) → (SP)	
	—	#n16	1 E 9 F	2	#	9	#<data> → -(SP)	
PUSHM	—	REGLIST	1 F 9 E	2	M	7+3n	REGLIST → -(SP) NOTE: A minimum of 2 registers must be specified and may be specified in any order.	— — — —
RET	—	—	5 6 D 7	1	—	9	(SP)+ → PC	— — — —
	—	Ax	5 0 Ax D 7	1	—	9	((Ax))+ → PC	
RETI	—	—	5 F D 7	1	—	12	(SP)+ → SR (SP)+ → PC	* * * *

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
ROL ROLC ROR RORC		see ASL (page 10)						
SUB SUBC		see ADD (page 8)						
TEST TESTN	.B [.W]	#n16,Ry	7 7 s c1 0 0 REGy	2	#	6	TEST: Src AND Dst	* * 0 0
	.B [.W]	#n16,Py	7 F s c1 0 1 PORTy	2	#	9	TESTN: Src AND .NOT.(Dst)	
	.B [.W]	#n16,(Ay)	7 7 s c1 0 1 0 Ay	2	#	9		

The following symbols are used to describe the state of the Status Register flags:
* Set according to result of operation.
0 Cleared.
1 Set.
— Not affected.
U Undefined.

Port Map

Port		Addr	Description	15	14	13	12	11	10	9	8
P0	PH0 PL0	$FC00	16 External I/O or Addr/Data Bus In Expanded Bus Mode								
P1	PH1 PL1	$FC02	16 Ext I/O, Ext Interrupts, Serial Port I/O, Bus Control	LB LDS	HB UDS	R/W	DS BGACK	DTACK	AS	BUSIN BG	BUSOUT BR
P2	PH2 PL2	$FC04	Reserved	← RESERVED →							
P3	PH3 PL3	$FC06	Serial Receive and Serial Transmit Buffers	RECEIVE BUFFER							
P4		$FC08	8 External I/O or Timer and Port 0 Handshake	TAO	TBO	TAI	TBI	STRH R/G	STRL	RDYH	RDYL
P5		$FC0A	Reserved	← RESERVED →							
P6		$FC0C	Reserved	← RESERVED →							
P7		$FC0E	Interrupt Latch Register	RES	NMI	SPARE	XI2I	STRLI	TAOI	TAII	STRHI
P8		$FC10	Interrupt Mask Register	RESERVED		SPARE	XI2M	STRLM	TAOM	TAIM	STRHM
P9		$FC12	Serial I/O Receive Control and Status Register	RE	IS	RW1	RW0	RC	SIS	RESERVED	
P10		$FC14	Serial I/O Transmit Control and Status Register	TE	AT	LM	TW1	TW0	TC	P/S	RES
P11		$FC16	Timer B Latch								
P12		$FC18	Timer A Low Latch								
P13		$FC1A	Timer A High Latch								
P14		$FC1C	Timer Control, Interrupt Edge Select	TEST	RESERVED		XI2C	RESERVED		TCOC	TAM1
P15		$FC1E	Port 0 Handshake mode, Bus Lock, Bus Segment Bits	SEG1	SEG0	BLCK	F/S	PM3	PM2	PM1	PM0
P16		$FC20	Port 0 Direction Control (DDR0)								
P17		$FC22	Port 1 Direction Control (DDR1)								
P18		$FC24	Serial I/O Mode and Sync Register	A/S	WL1	WL0	ST	PAR1	PAR0	TCO	WS
P19		$FC26	Timer C Latch								

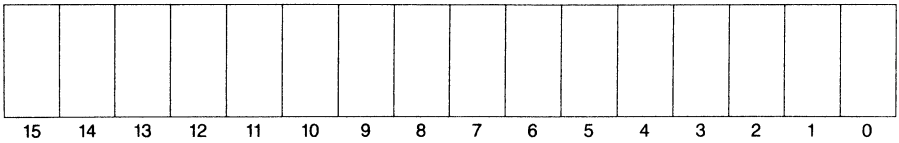
Port Map								
7	6	5	4	3	2	1	0	Initial Condition
								—
I/O	XI2	XI1	XI0	SI	RCLK	TCLK	SO	—
← RESERVED →								—
TRANSMIT BUFFER								—
← RESERVED →								High Byte 00----00
← RESERVED →								—
← RESERVED →								—
RSCI	RNI	XI1I	TBOI	TBII	XI0I	XMTI	TCI	$0000
RSCM	RNM	XI1M	TBOM	TBIM	XI0M	XMTM	TCM	$0000
BF	OE	PE	FE	SF/AF	← RESERVED →			$0000
BE	UE	END	← RESERVED →					$00A0
								—
								—
								—
TAM0	TAE	TAIC	TAOC	TBM1	TBM0	TBE	TBIC	$0000
← RESERVED →								Mode pin: VCC-$0C00 CLKOUT-$0E00 GND-$0F00
								$0000
								$0000
SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	$0000
								—

NOTE:
When a reserved bit is read, it is read as a zero.

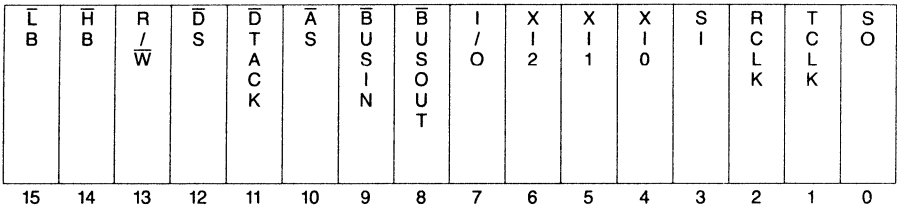
PORT DESCRIPTIONS

Important note: All port bits that are not explicitly defined are reserved for possible use in future MK68200 family members. If reserved bits are written, they should be written with zero values. When reserved bits are read, they are read as zeros.

PORT 0 — 16 External I/O Bits; read/write
$FC00 (used as address/data lines in external bus configurations)

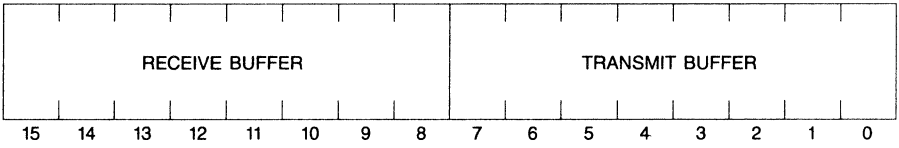


PORT 1 —16 External I/O Bits, including Interrupt, Serial I/O,
$FC02 and Bus Control (shown for GP bus); read/write



PORT 2 —Reserved

PORT 3 —Serial I/O:Transmit Buffer (Low byte; read/write)
$FC06 Receive Buffer (High Byte; read only)



PORT 4 —8 External I/O Pins (Timer and Port 0 Handshake Lines);
$FC08 Inputs: read only
Outputs: read/write

TAO	TBO	TAI	TBI	STRH	STRL	RDYH	RDYL	(reserved)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Note: STRH is also R/$\overline{G}$ in the expanded bus modes.

PORT 5 —Reserved

PORT 6 —Reserved

PORT 7 —Interrupt Latch Register; read/write
$FC0E

res	NMI	SPARE	XI2I	STRLI	TAOI	TAII	STRHI	RSCI	RNI	XI1I	TBOI	TBII	XI0I	XMTI	TCI
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PORT 8 —Interrupt Mask Register; read/write
$FC10

(res)	SPARE	XI2M	STRLM	TAOM	TAIM	STRHM	RSCM	RNM	XI1M	TBOM	TBIM	XI0M	XMTM	TCM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INTERRUPT AND RESET VECTORS

VECTOR NUMBER	NAME	MNEMONIC	VECTOR LOCATION	
0	RESET	RESET	$0000	
1	NON-MASKABLE INTERRUPT	NMI	$0002	LEVEL 2
2	SPARE	SPARE	$0004	
3	EXTERNAL INTERRUPT 2	XI2	$0006	
4	STROBE LOW	STRL	$0008	
5	TIMER A OUTPUT	TAO	$000A	
6	TIMER A INPUT	TAI	$000C	
7	STROBE HIGH	STRH	$000E	
8	RECEIVE SPECIAL CONDITION	RSC	$0010	LEVEL 1
9	RECEIVE NORMAL	RN	$0012	
A	EXTERNAL INTERRUPT 1	XI1	$0014	
B	TIMER B OUTPUT	TBO	$0016	
C	TIMER B INPUT	TBI	$0018	
D	EXTERNAL INTERRUPT 0	XI0	$001A	
E	TRANSMIT	XMT	$001C	
F	TIMER C	TC	$001E	

NOTE: Reset and Interrupt sources are listed in order of decreasing priority with RESET having the highest priority.

PORT 9 —Serial I/O Receive Control and Status Register;
$FC12 High byte: control register; read/write
Low byte: status register; read only

RE	IS	RW1	RW0	RC	SIS	(res)		BF	OE	PE	FE	SF/AF	(res)		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit Descriptions:

RE
(Receiver Enable control)

0 = Disabled; all status flags cleared.
1 = Enabled.

IS
(Ignore Syncs control)

0 = Disabled; interrupts may occur on all characters received.
1 = Enabled; interrupts cannot occur on sync characters received after the sync match is found.

RW1, RW0
(Receiver Wake-up control)

The receiver wake-up control bits operate as follows.

MODE	RW1	RW0	APPENDED WAKE-UP	BUFFER LOADED	INTERRUPT GENERATED
No Wake up	0	0	no	any character	RN
Wake-up on Any Character	0	1	yes	any character	RN
Wake-up on Address Match	1	0	yes	address match	RSC
Wake-up on Any Address	1	1	yes	any address	RSC

RC
(Receive Clock control)

0 = Selects external receive clock applied on RCLK.
1 = Selects internal clock from the on-chip baud rate generator (Timer C) for the receive clock.

This bit is ignored when either the TCO bit or the LM (Loopback Mode) bit is set.

SIS
(Single Interrupt Select control)

0 = Separate vectors are generated for the Receive Normal and the Receive Special Condition interrupts.
1 = The Receive Normal vector is generated for all receive character interrupts.

BF
(Buffer Full status)

0 = Receive data buffer empty; cleared when receive buffer is read.
1 = Receive buffer full; set when an incoming word is loaded into the receive data buffer.

OE
(Overrun Error status)

0 = No overrun error; cleared when the status register is read.
1 = Overrun error; set when a new word has been received and the previous word has not been read from the receive data buffer.

PE
(Parity Error status)

0 = No Parity Error; cleared when the status register is read.
1 = Parity Error; set when a parity error has been detected on an incoming character in the data stream.

FE
(Frame Error status)

0 = No frame error; cleared when the status register is read.
1 = Frame error; set when a word is transferred to the receive data and no stop bit has been recognized.

This flag applies to async formats only.

SF/AF
(Sync Found or Address Found status)

This flag is used for both sync character match conditions and address found conditions in some wake-up modes as follows.

	MODES			
SS	**IS**	**RW1**	**RW0**	**CONDITIONS THAT SET SF/AF**
1	x	x	x	Sync Found on any bit boundry
0	1	x	x	unaffected
0	0	0	0	unaffected
0	0	1	1	Any Address
0	0	1	0	Address Match
0	0	0	1	unaffected

PORT 10—Serial I/O Transmit Control and Status Register
$FC14 High byte: control register; read/write
Low byte: status register; read only

TE	AT	LM	TW1	TW0	TC	P/$\overline{S}$	res	BE	UE	END	(reserved)				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit Descriptions:

TE
(Transmitter Enable control)

0 = Disable the transmitter; any word being shifted out will continue until completion.
1 = Enable the transmitter.

AT
(Automatic Turn Around control)

0 = No effect on TE or RE.
1 = Causes RE to be set to a "1" and TE to be set to a "0" automatically at the end of a transmission.

LM
(Loopback Mode control)

0 = Disables loopback mode.
1 = Causes the transmitter output to be internally connected to receiver input. Also causes Timer C to be used for both the transmit and receive clocks regardless of the state of TC, RC, TCO, and TCOC.

TW1, TW0
(Transmit Wake-up control)

These bits provide control for wake-up operation as follows.

TW1	TW0	OPERATION
1	0	Transmit Data
1	1	Transmit Address
0	X	No Wake-up

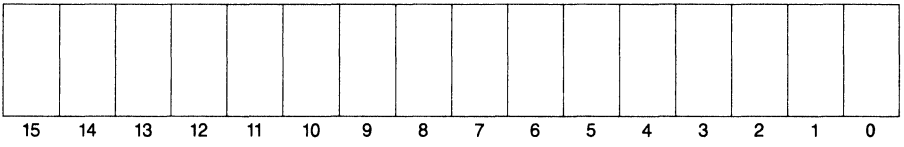
TC
(Transmit Clock control)

0 = Selects the external clock signal applied on TCLK for the transmit clock.
1 = Selects the internal baud rate generator output (Timer C) for the transmit clock.

This bit is ignored if either the TCO bit or the LM bit is set.

P/$\overline{S}$ (Previous/Sync control)	0 = Selects continuous transmission of the contents of the sync character register in the synchronous mode when there is no data to transmit. 1 = Selects continuous transmission of the transmit data buffer in synchronous mode when there is no data to transmit.
BE (Buffer Empty status)	0 = Transmit Buffer is full; reset to this condition after the transmit buffer is reloaded. 1 = Transmit Buffer is empty; set to this condition after the transmit buffer contents are transferred to the output shift register.
UE (Underrun Error status)	0 = No underrun error; cleared following a read of the transmit buffer. 1 = Underrun error; set only in the synchronous mode when the last word has been shifted out and transmit buffer has not been reloaded.
END (End of Transmission status)	0 = No end of transmission; cleared by enabling the transmitter. 1 = End of transmission detected; set when the transmitter is disabled and the last character has been shifted out.

PORT 11— Timer B Latch; read gets counter value;
$FC16 write goes to latch



PORT 12—Timer A, Low Latch; read gets counter or latch value;
$FC18 write goes to latch

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PORT 13—Timer A, High Latch; read gets counter or latch value;
$FC1A write goes to latch

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PORT 14—Timer Control, Interrupt Edge Select; read/write
$FC1C

TEST	(res)	XI2C	(res)	TCOC	TAM1	TAM0	TAE	TAIC	TAOC	TBM1	TBM0	TBE	TBIC		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit Descriptions:

TEST
(Test mode control)
0 = Normal operation; cleared on user control and on reset.
1 = Selects test mode; not to be used during normal operation.

XI2C
(External Interrupt 2 Control)
0 = Interrupt on falling edge of XI2.
1 = Interrupt on rising edge of XI2.

TCOC
(Timer C Output Control)
0 = When TCO = 1, TCLK is selected for use as a general purpose I/O pin.
1 = When TCO = 1, TCLK is selected for use as an output for Timer C.

TCOC has no effect when TCO = 0.

TAM1, TAM0
(Timer A Mode control)

These bits select the operating mode of Timer A as follows.

TAM1	TAM0	MODE
0	0	Interval
0	1	Event
1	0	Pulse/period 1
1	1	Pulse/period 2

TAE
(Timer A Enable control)

0= Disables Timer A; all Timer A operations are inhibited, and the timer counter is initialized.
1= Enables Timer A; the timer begins operation as defined by the other Timer A control bits.

TAIC
(Timer A Input Control)

0= Selects a negative transition as the active edge for TAI.
1= Selects a positive transition as the active edge for TAI.

TAOC
(Timer A Output Control)

0= Selects TAO as a general purpose output pin.
1= Selects TAO as an ouput pin associated with Timer A; TAO is initialized low when TAOC is a one and TAE is zero.

TBM1, TBM0
(Timer B Mode control)

These bits select the operating mode of Timer B as described below.

TBM1	TBM0	MODE
0	0	Interval 0 (TBO is not used)
0	1	Interval 1 (TBO is used)
1	0	Retriggerable one-shot
1	1	Non-retriggerable one-shot

TBE
(Timer B Enable control)

0= Disables Timer B; all operations are inhibited, and the timer counter is initialized.
1= Enables Timer B; the timer begins operation as defined by the other Timer B control bits.

TBIC
(Timer B Input Control)

0= Selects a negative transition as active on TBI.
1= Selects a positive transition as active on TBI.

PORT 15—Port 0 Handshake Mode, Fast/Standard, Bus Lock, and Bus
$FC1E Segment Bits; read/write

SEG1	SEG0	BLCK	F/$\overline{S}$	PM3	PM2	PM1	PM0	(reserved)
15	14	13	12	11	10	9	8	7 6 5 4 3 2 1 0

Bit Descriptions:

SEG1, SEG0
(Segment bits)

Used in the expanded bus mode when a reference is made to the DMA window. The contents of SEG1 and SEG0 are then output on pins AD15 and AD14, respectively.

BLCK
(Bus Lock control)

0 = Disables the bus lock function.
1 = Enables the bus lock function.

F/$\overline{S}$
(Fast/$\overline{\text{Standard}}$ timing control)

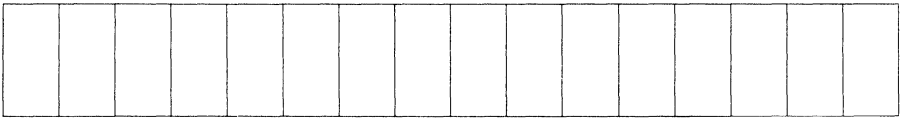
0 = Selects standard timing of read/write cycles on the external bus (4 clock periods).
1 = Selects fast timing of read/write cycles on the external bus (3 clock periods.)

PM0, PM1, PM2, PM3
(Port Mode control)

These bits allow the user to select one of eight different handshaking modes.

PM3	PM2	PM1	PM0	HIGH HANDSHAKE	LOW HANDSHAKE
0	0	0	0	Inactive	PL0 or P0 output
0	0	0	1	PH0 output	PL0 output
0	0	1	0	Inactive	PL0 input
0	0	1	1	PH0 input	PL0 input
0	1	0	0	PH0 input	PL0 ouput
0	1	0	1	Inactive	P0 input (word only)
0	1	1	0	PL0 output	PL0 input (bidirectional)
0	1	1	1	P0 output	P0 input (bidirectional)
—	—	—	—		
1	1	0	0	Inactive	Inactive
—	—	—	—		
1	1	1	0	Full Expansion (62K external bus)	
1	1	1	1	Partial Expansion (16K external bus)	

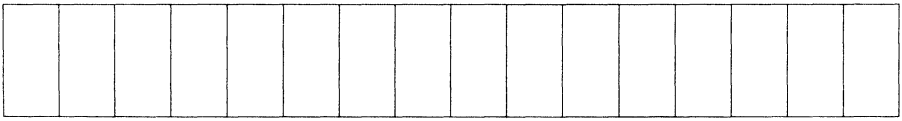
PORT 16—Port 0 Data Direction Control (DDR0); read/write
$FC20



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

0—Corresponding Port 0 bit is input.
1—Corresponding Port 0 bit is output.

PORT 17—Port 1 Data Direction Control (DDR1); read/write
$FC22



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

0—Corresponding Port 1 bit is input.
1—Corresponding Port 1 bit is output.

PORT 18—Serial I/O Mode and Sync Register; read/write
$FC24

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A/$\overline{S}$	WL1	WL0	ST	PAR1	PAR0	TCO	WS	SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0

Bit Descriptions:

A/$\overline{S}$
(Asynchronous/$\overline{\text{Synchronous}}$ mode control)

0 = Selects synchronous operation for the serial port; transmit and receive clocks are divided by 1.
1 = Selects asynchronous operation for serial port; transmit and receive clocks are divided by 16.

WL1, WL0
(Word Length control)

These two bits select the length of the data word as follows.

WL1	WL0	Word Length
0	0	8 bits
0	1	7 bits
1	0	6 bits
1	1	5 bits

ST
(Stop Bit control)

This bit is only used in the asynchronous mode. It selects the number of stop bits transmitted.

ST	Number of Stop Bits
0	1
1	2

PAR1, PAR0
(Parity control)

These two bits provide parity control for both the synchronous and asynchronous modes.

PAR1	PAR0	Parity
0	0	no parity
0	1	fixed "0" parity
1	0	odd parity
1	1	even parity

Note that even parity is defined such that the sum of the data and parity bits is even.

TCO
(Timer C Output mode control)

0= Disables Timer C output mode.
1= Enables Timer C output mode; disables Timer C's use as a baud rate generator when LM = 0; causes transmit and receive clocks to be internally connected to RCLK so that TCLK may be used either as general purpose I/O or as an output for Timer C.

WS

(Wake-up Sense)

The following table lists the effects of
the WS bit.

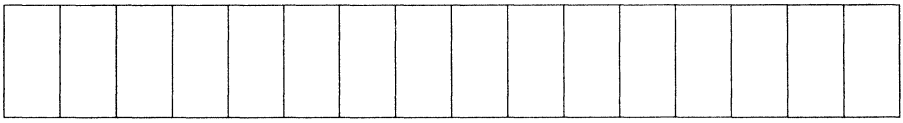
WS	Wake-up bit	Meaning
0	0	Address Word
0	1	Data Word
1	0	Data Word
1	1	Address Word

SYNC7-SYNC0

(Sync character bits)

These eight bits are used to store the
sync character or the device address for the wake-up mode.

PORT 19—Timer C Latch; read gets counter, write goes to latch and counter
$FC26



ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Temperature Under Bias . . . . . −25°C to +100°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin with Respect to Ground . . . . . −0.3 V to +7 V
Power Dissipation . . . . . 1.5 W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

MK68200 DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 V ± 5%, T_a = 0° to 70°C)

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
V_{IL}	Input low voltage; all inputs	−0.3	0.8	V	
V_{IH}	Input high voltage; all inputs	2.0	V_{CC}	V	
V_{OL}	Output low voltage; all outputs		0.4	V	I_{OL} = 2.0 mA
V_{OH}	Output high voltage; all outputs	2.4		V	I_{OH} = −250μA
I_{CC}	Input power supply current		220	mA	Outputs Open
I_{LI}	Input leakage current		±10	μA	V_{IN} = 0 to V_{CC}
I_{LO}	Three-state output leakage current in float		±10	μA	V_{OUT} = 0.4 V to V_{CC}

CAPACITANCE

T_a = 25°C, f = 12 MHz with unmeasured pins returned to ground.

SYMBOL	PARAMETER	MAX	UNIT	TEST CONDITION
C_{IN}	Input Capacitance	10	pF	Unmeasured pins returned to ground
C_{OUT}	Three-state Output Capacitance	10	pF	

MK68200 AC ELECTRICAL SPECIFICATIONS

T_a = 0°C to 70°C, V_{CC} = +5V ±5% unless otherwise specified.
AC measurements are referenced from minimum V_{IH} or maximum V_{IL} for inputs and from minimum V_{OH} or maximum V_{OL} for outputs.

NO.	DESCRIPTION	4 MHz		6MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX		
1	$\overline{\text{RESET}}$ low time	20		20		state times	1
2	CLK 1 width high (external clock input)	45		30		ns	
3	CLK 1 width low (external clock input)	45		30		ns	
4	CLK 1 period (external clock input)	125	1000	83	1000	ns	
5	Crystal input frequency	1.000	8.000	1.000	12.000	MHz	
6	Clock Period (PHI 1)	250		167		ns	
7	PHI 1 low to PHI 1 high	125		83		ns	
8	PHI 1 high to PHI 1 low	125		83		ns	
9	PHI 1 low to CLKOUT low		40		27	ns	
10	PHI 1 high to CLKOUT high		40		27	ns	

MK68200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (UPC, GP, AND PRIVATE BUSES)

NO.	DESCRIPTION	4MHz		6MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX		
11	PHI 1 low to R/$\overline{\text{W}}$, $\overline{\text{HB}}$, or $\overline{\text{LB}}$ valid		115		76	ns	2
12	PHI 1 high to $\overline{\text{AS}}$ low		115		76	ns	2
13	PHI 1 low to address valid		115		76	ns	2
14	$\overline{\text{AS}}$ low to address invalid	70		50		ns	2
15	PHI 1 low to tri-state address		90		60	ns	2
16	Tri-state address to $\overline{\text{DS}}$, $\overline{\text{LDS}}$, or $\overline{\text{UDS}}$ starting low (fast cycle)	10		10		ns	2
17	PHI 1 low to $\overline{\text{DS}}$, $\overline{\text{LDS}}$, or $\overline{\text{UDS}}$ low (fast cycle)		165		110	ns	2
18	PHI 1 low to data out valid during write		115		76	ns	2
19	PHI 1 low to R/$\overline{\text{W}}$, $\overline{\text{HB}}$, $\overline{\text{LB}}$ invalid	0		0		ns	2
20	PHI 1 low to address/data bus driven	0		0		ns	2
21	$\overline{\text{AS}}$ low to $\overline{\text{DS}}$, $\overline{\text{LDS}}$, or $\overline{\text{UDS}}$ starting low (fast cycle)	100	225	70	150	ns	2

MK68200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (UPC AND GP BUSES)

NO.	DESCRIPTION	4 MHz		6MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX		
22	Tri-state address to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ starting low (standard cycle)	135		90		ns	
23	PHI 1 high to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ low (standard cycle)		165		110	ns	2
24	Valid Data Setup to PHI 1 low	10		5		ns	2
25	$\overline{AS}$ low to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ starting low (standard cycle)	225	350	150	230	ns	2
26	R/$\overline{W}$, $\overline{HB}$, or $\overline{LB}$ valid to $\overline{AS}$ starting low	60		60		ns	
27	Address valid to $\overline{AS}$ starting low	60		60		ns	
28	Input data hold time from PHI 1 low	45		30		ns	
29	Input data hold time from $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ high	0		0		ns	
30	PHI 1 low to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ high		180		120	ns	
31	$\overline{DTACK}$ low setup to PHI 1 high	15		10		ns	
32	$\overline{LDS}$, $\overline{UDS}$, or $\overline{DS}$ high to $\overline{DTACK}$ high (hold time)	−30		−30		ns	
33	$\overline{LDS}$, $\overline{UDS}$, or $\overline{DS}$ pulse width	240		150		ns	
34	PHI 1 high to $\overline{AS}$ high		90		60	ns	
35	PHI 1 low to data out invalid	0		0		ns	
36	$\overline{AS}$ inactive	235		150		ns	
37	$\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ high to data out invalid	180		110		ns	
38	$\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ high to $\overline{AS}$ high	5		5		ns	

MK68200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (UPC BUS)

NO.	DESCRIPTION	4 MHz		6 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX		
39	$\overline{BGACK}$ low to $\overline{BR}$ high	100	450	100	300	ns	
40	$\overline{BG}$ low to $\overline{BGACK}$ low	50	600	50	400	ns	
41	$\overline{BGACK}$, $\overline{AS}$, $\overline{DTACK}$, inactive to $\overline{BGACK}$ low; $\overline{BG}$ already low	0	600	0	400	ns	
42	$\overline{BGACK}$ low to $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, or address/data bus driven	40	135	40	90	ns	
43	$\overline{AS}$, $\overline{LDS}$, $\overline{UDS}$ or address/data bus tri-state to $\overline{BGACK}$ high	0	180	0	120	ns	

MK68211 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (GP BUS)

NO.	DESCRIPTION	4 MHz		6 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX		
44	Tri-state $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$ to $\overline{BUSOUT}$ low (bus grantor, fast cycle, (no wait states)	175		100		ns	
45	$\overline{BUSIN}$ low to $\overline{BUSOUT}$ low (bus grantor, fast cycle, no wait states)		1900		1200	ns	
46	$\overline{BUSOUT}$ high to $\overline{AS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$ driven (bus grantor)	15		15		ns	
47	$\overline{BUSIN}$ high to $\overline{BUSOUT}$ high (bus grantor)	520	900	300	600	ns	
48	Tri-state address/data bus to $\overline{BUSOUT}$ low (bus grantor)	70		70		ns	
49	$\overline{BUSOUT}$ high to address/data bus driven (bus grantor)	50		50		ns	
50	$\overline{BUSOUT}$ low to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$ driven (bus requestor, $\overline{BUSIN}$ low)	240		150		ns	
51	$\overline{BUSIN}$ low to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$ driven (bus requestor, $\overline{BUSOUT}$ low)	270	650	180	500	ns	
52	Tri-state $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$, to $\overline{BUSOUT}$ high (bus requestor)	180		100		ns	
53	$\overline{BUSOUT}$ high to $\overline{BUSIN}$ high (bus requestor)		530		400	ns	
54	$\overline{BUSIN}$ low to address/data bus driven (bus requestor)	350		250		ns	
55	Tri-state address/data bus to $\overline{BUSOUT}$ high (bus requestor)	100		65		ns	

MK68E200 BUS AC ELECTRICAL SPECIFICATIONS (PRIVATE BUS)

NO.	DESCRIPTION	4 MHz		6 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX		
56	Valid Data Setup to PHI 1 low	30		20		ns	
57	PBR/$\overline{W}$ valid to $\overline{PBAS}$ starting low	40		40		ns	
58	Address valid to $\overline{PBAS}$ starting low	35		35		ns	
59	Input data hold time from PHI 1 low	0		0		ns	
60	Input data hold time from $\overline{PBDS}$ high	−25		−25		ns	

MK68E200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (PRIVATE BUS) (Cont.)

NO.	DESCRIPTION	4 MHz		6 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX		
61	PHI 1 low to $\overline{\text{PBDS}}$ high		160		105	ns	
62	$\overline{\text{PBDTACK}}$ low setup to PHI 1 high	20		15		ns	
63	$\overline{\text{PBDS}}$ high to $\overline{\text{PBDTACK}}$ high (hold time)	−15		−15		ns	
64	$\overline{\text{PBDS}}$ pulse width	190		125		ns	
65	PHI 1 high to $\overline{\text{PBAS}}$ high		115		75	ns	
66	PHI 1 low to data out invalid	10		10		ns	
67	$\overline{\text{PBAS}}$ inactive	200		135		ns	
68	$\overline{\text{PBDS}}$ high to data out invalid	200		135		ns	
69	$\overline{\text{PBDS}}$ high to $\overline{\text{PBAS}}$ high	15		15		ns	

MK68200 INPUT/OUTPUT AC ELECTRICAL CHARACTERISTICS

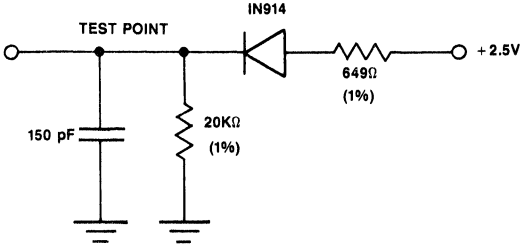
NO.	DESCRIPTION		4 MHz		6 MHz		UNITS	NOTES
			MIN	MAX	MIN	MAX		
70	Active and inactive pulse times	For XI2, XI1, STRH, STRL, TAI, TBI, $\overline{\text{NMI}}$	5		5		state times	1
		For XI0	3		3			
71	Input data setup to falling edge of STRH, STRL		15		10		ns	
72	Input data hold from the falling edge of STRH, STRL		60		40		ns	
73	RDYH, RDYL low time		1	3	1	3	state times	1
74	Delay from STRH, STRL high to RDYH, RDYL low			110		75	ns	
75	Delay from data valid to RDYH, RDYL high (output mode)			3		3	state times	1
76	Delay from STRH high to data out (bidirectional mode)			90		60	ns	
77	Port 0 data hold time from STRH low (bidirectional mode)		25		20		ns	
78	Delay to Port 0 float from STRH low (bidirectional mode)			85		55	ns	
79	TCLK,RCLK period (asynchronous)	as input	.250	DC	.167	DC	μs	
		as output	.500	DC	.334	DC		
	TCLK,RCLK period (synchronous)		1.0	DC	.667	DC		
80	TCLK, RCLK width low	as input	1	DC	1	DC	state	1
		as output	2	DC	2	DC	times	
81	TCLK, RCLK width high	as input	1	DC	1	DC	state	1
		as output	2	DC	2	DC	times	

MK68200 INPUT/OUTPUT AC ELECTRICAL SPECIFICATIONS

NO.	DESCRIPTION		4 MHz		6 MHz		UNITS	NOTES
			MIN	MAX	MIN	MAX		
82	TCLK low to SO delay (sync mode)	TCLK as input	330		220		ns	
		TCLK as output	75		50			
83	SI to RCLK high setup time (sync mode)	RCLK as input	30		20		ns	
		RCLK as output	180		120			
84	SI hold time from RCLK high (sync mode)	RCLK as input	45		30		ns	
		RCLK as output	0		0			

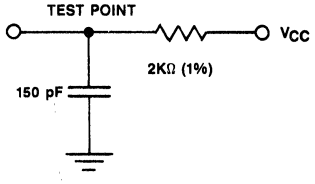
NOTES
1. One state time is equal to one-half of the instruction clock (PHI 1) period.
2. For the private bus case, the signals referenced apply to the equivalent private bus signals.

LOAD 1



TEST LOAD 1 IS APPLICABLE TO ALL PINS EXCEPT P1-12 AND P1-8.

LOAD 2



TEST LOAD 2 IS APPLICABLE TO P1-12 AND P1-8.

Figure 21. Output Test Load

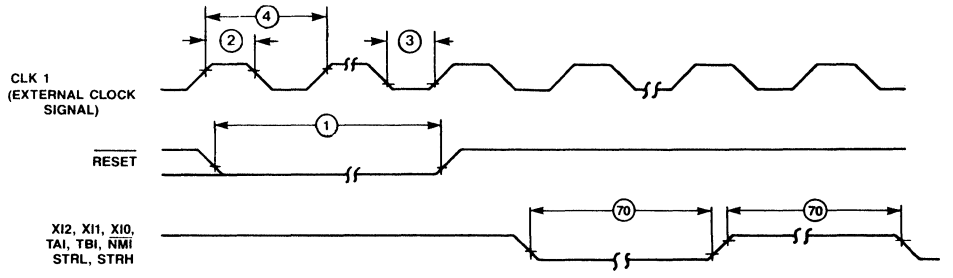


Figure 22. MK68200 AC Timing

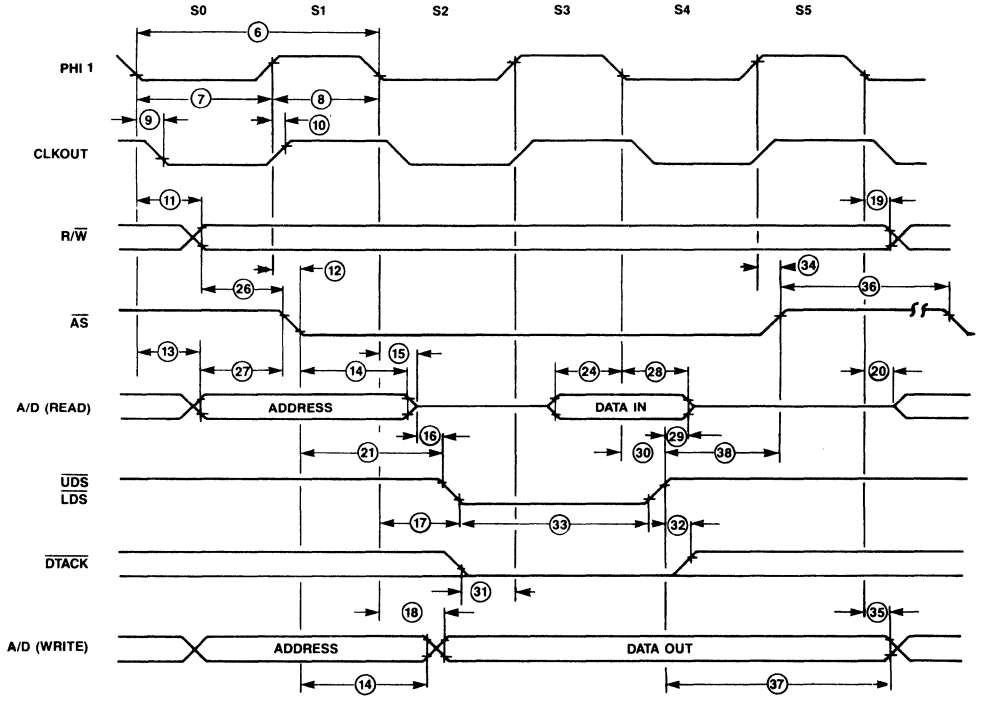


Figure 23. MK68201 UPC Bus Timing (Fast Cycle)

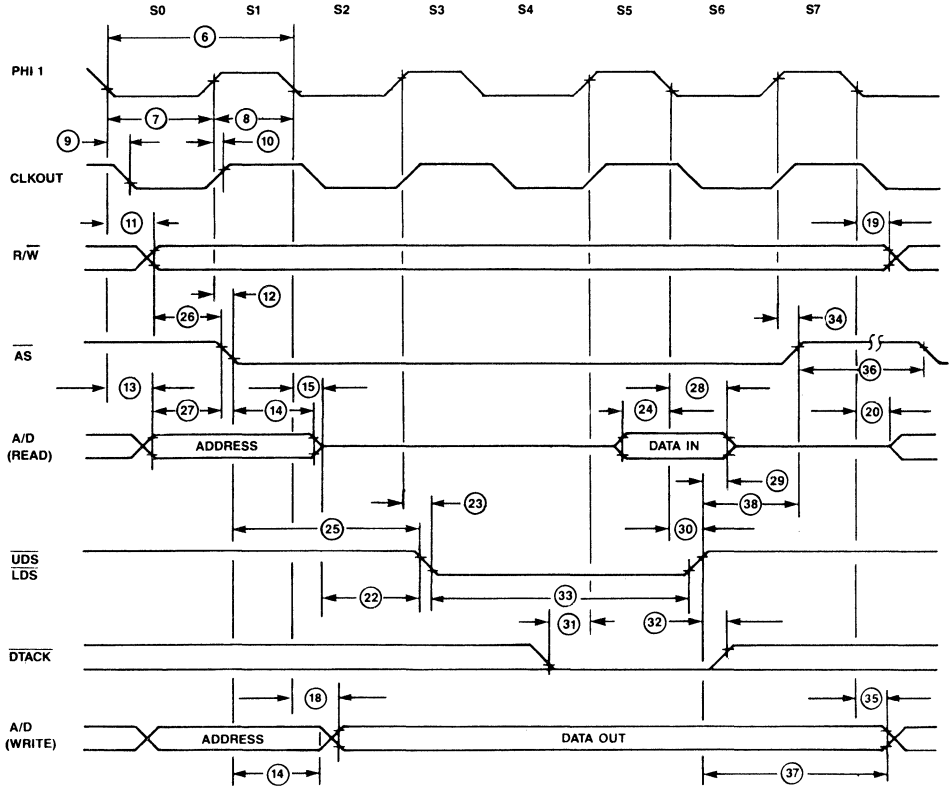


Figure 24. MK68201 UPC Bus Timing (Standard Cycle)

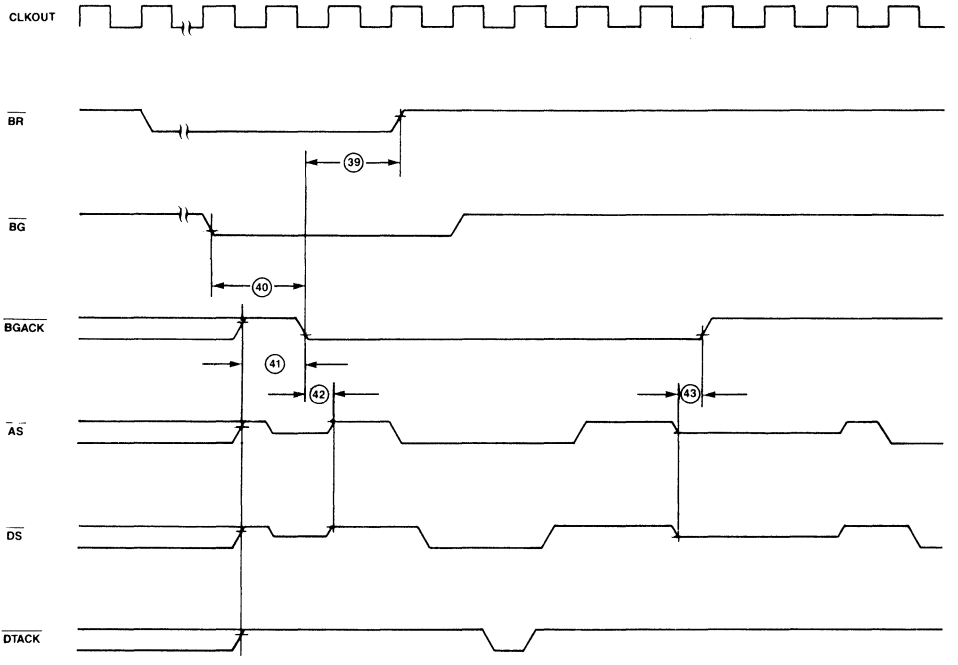


Figure 25. MK68201 UPC Bus Arbitration Timing

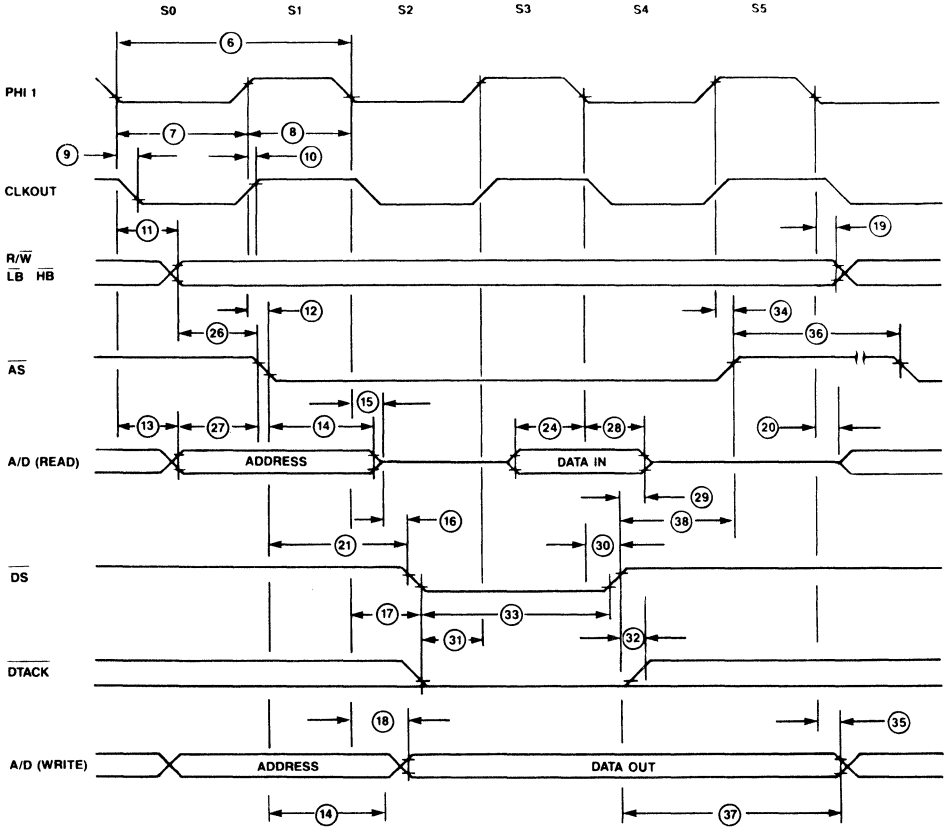


Figure 26. MK68211 GP Bus Timing (Fast Cycle)

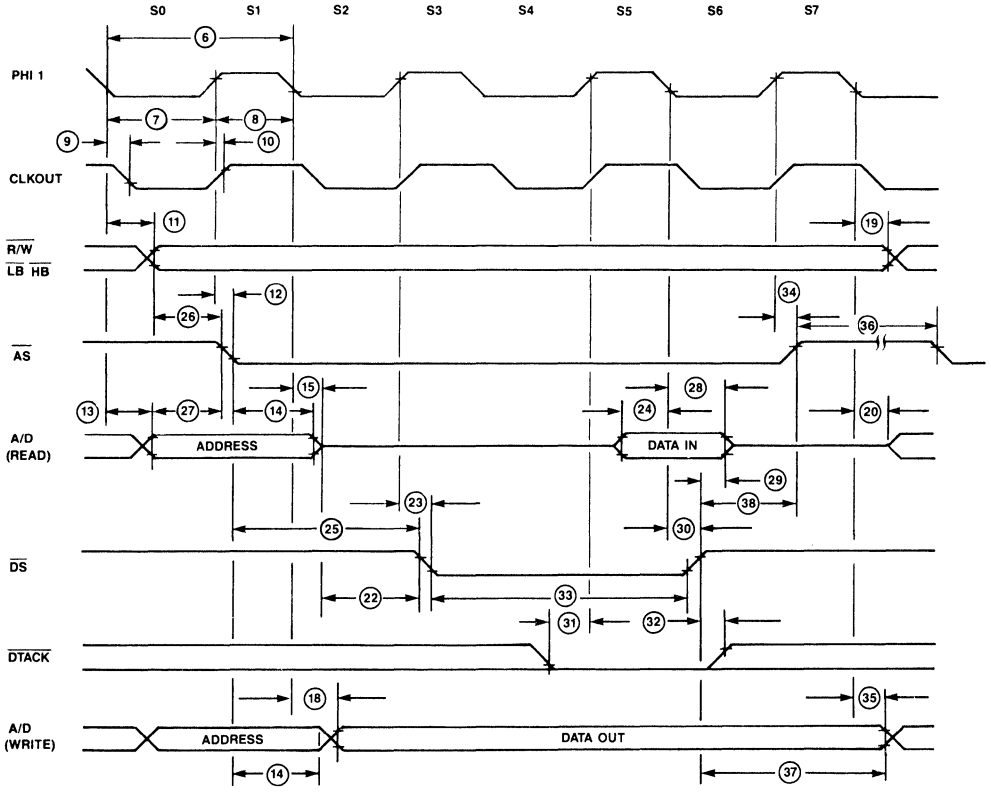


Figure 27. MK68211 GP Bus Timing (Standard Cycle)

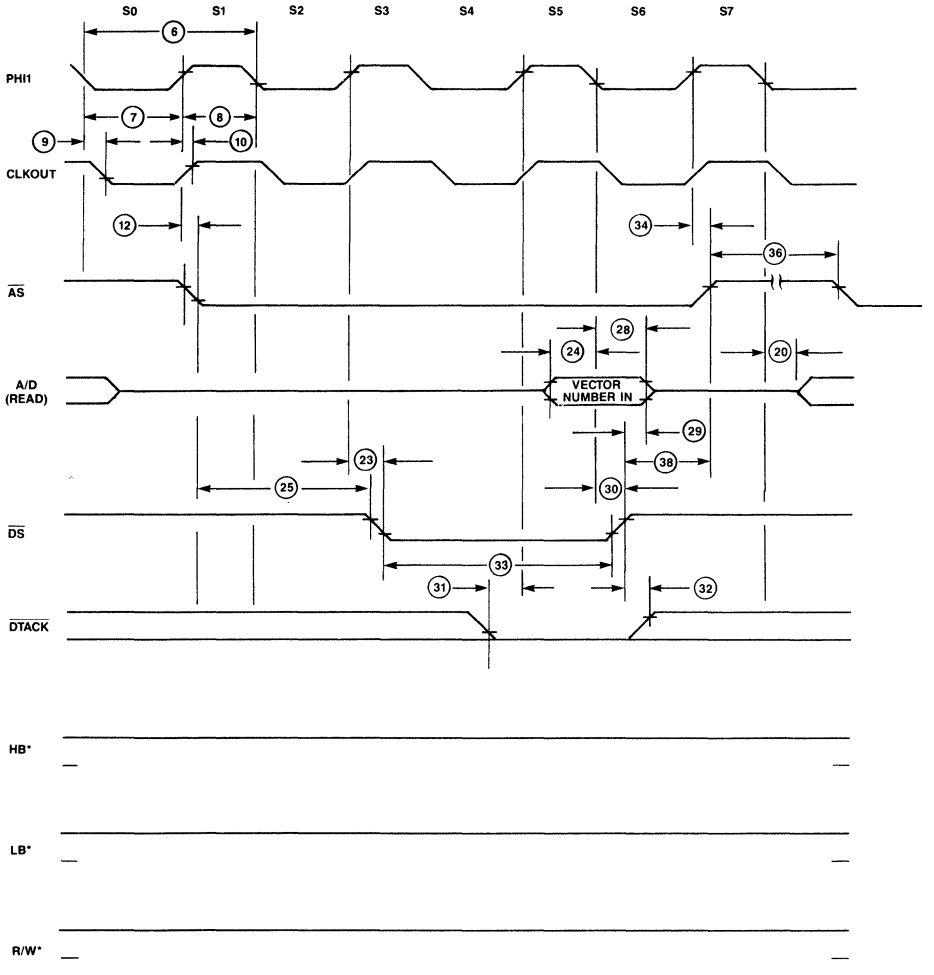


Figure 28. MK68211 GP Bus Timing (Interrupt Acknowledge Timing)

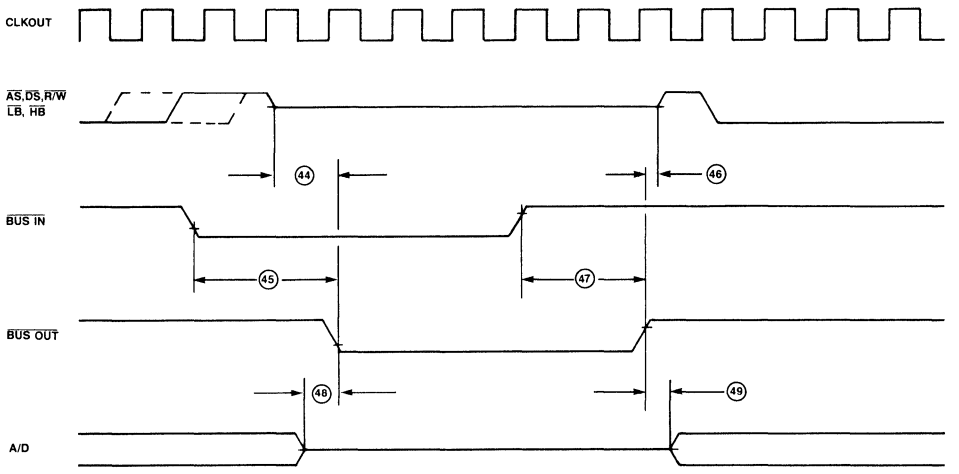


Figure 29. MK68211 GP Bus Arbitration Timing (Bus Grantor)

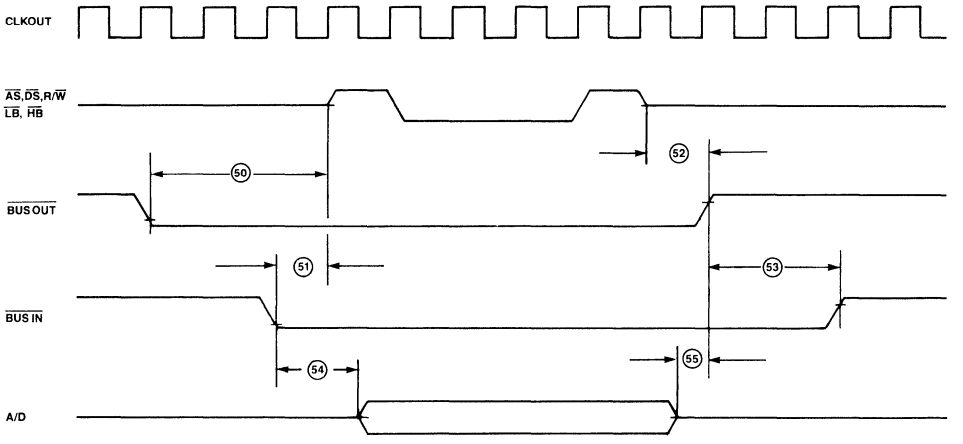


Figure 30. MK68211 GP Bus Arbitration Timing (Bus Requestor)

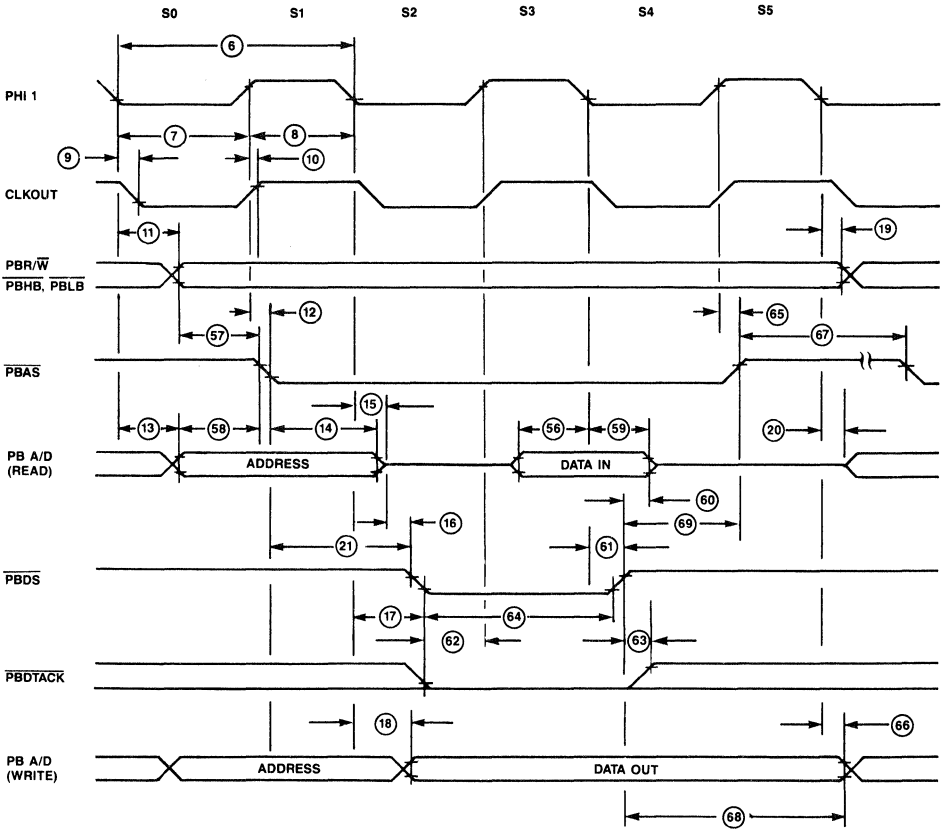


Figure 31. MK68200 Private Bus Timing (Fast Cycle)

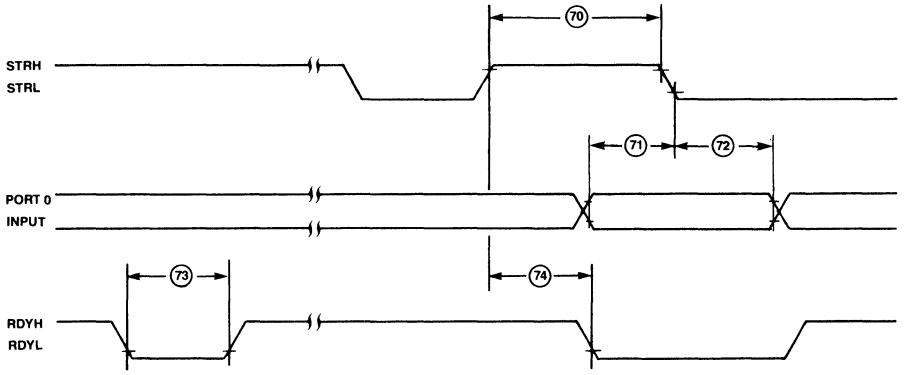


Figure 32. Input/Output AC Timing (Data Input)

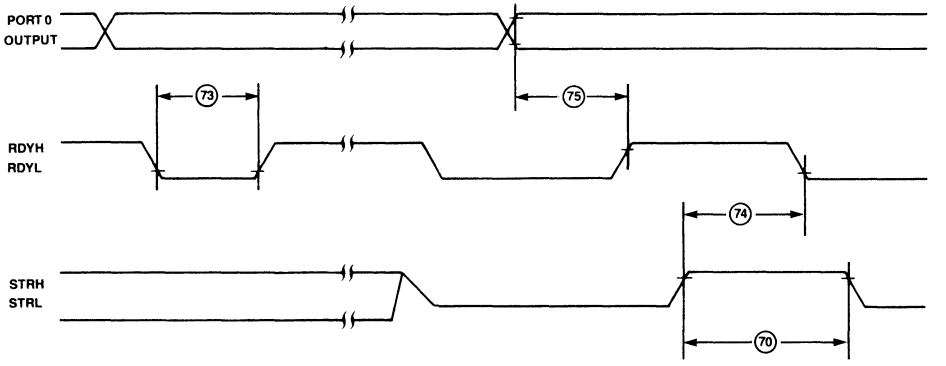


Figure 33. Input/Output AC Timing (Data Output)

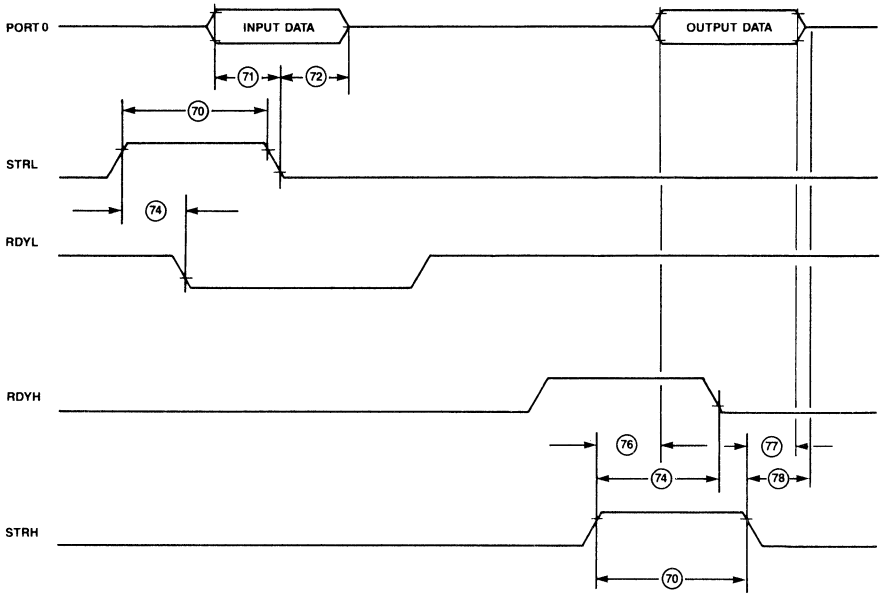


Figure 34. Input/Output AC Timing (Bidirectional I/O)

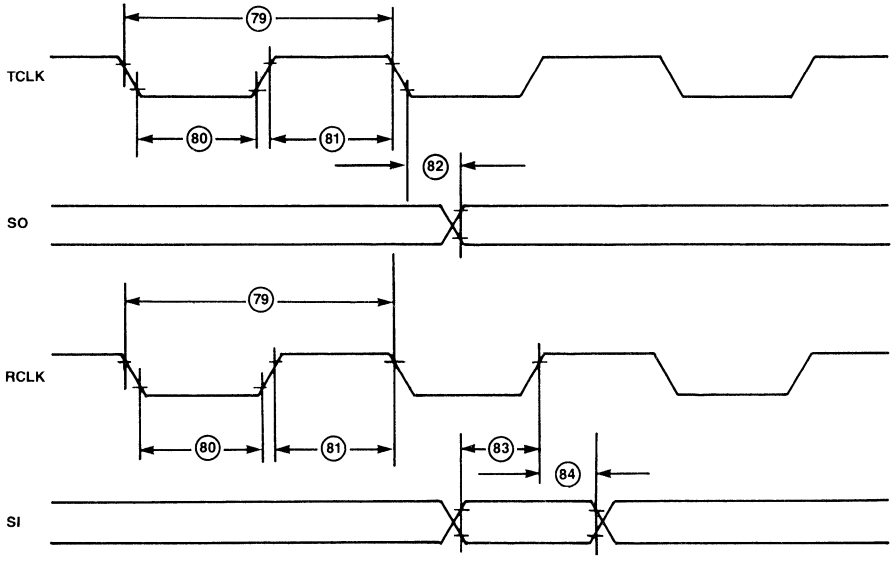


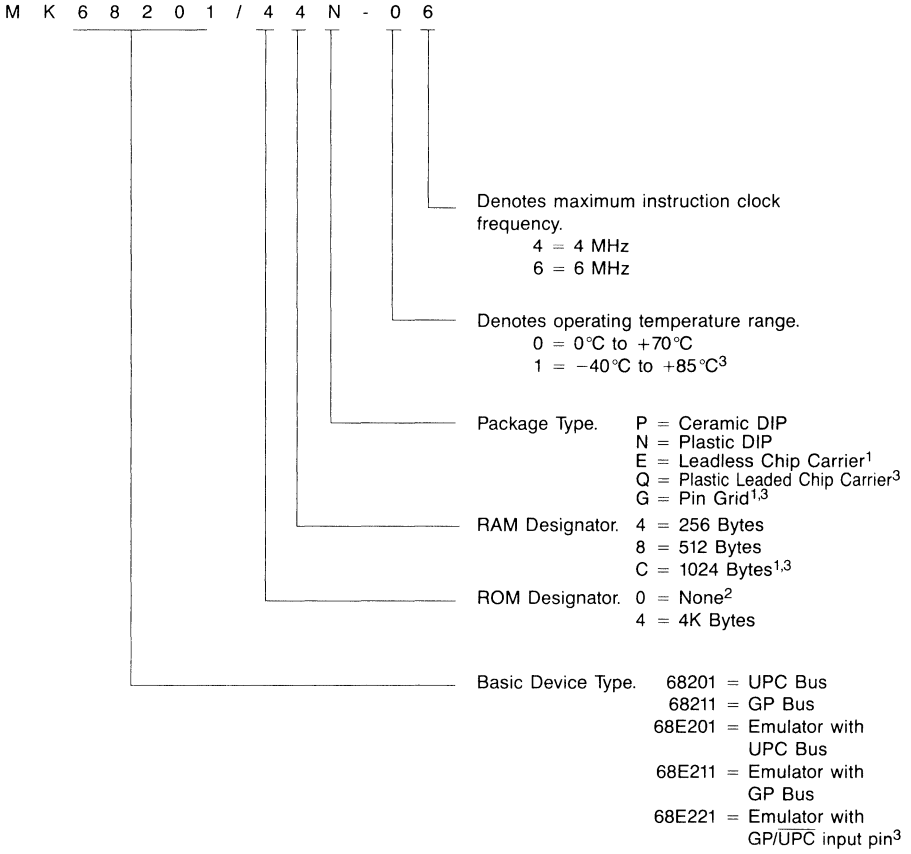
Figure 35. Input/Output AC Timing (Serial I/O)

PART NUMBERING INFORMATION

There are two types of part numbers for the 68200 family of devices. The generic part number describes the basic device type, the amount of ROM and RAM, the desired package type, temperature range, power supply tolerance, and expandable bus interface type. The device order number indicates the specific mask set Mostek will use to manufacture the device, along with package type, speed grade and temperature range.

Generic Part Number

An example of the generic part number is shown below:

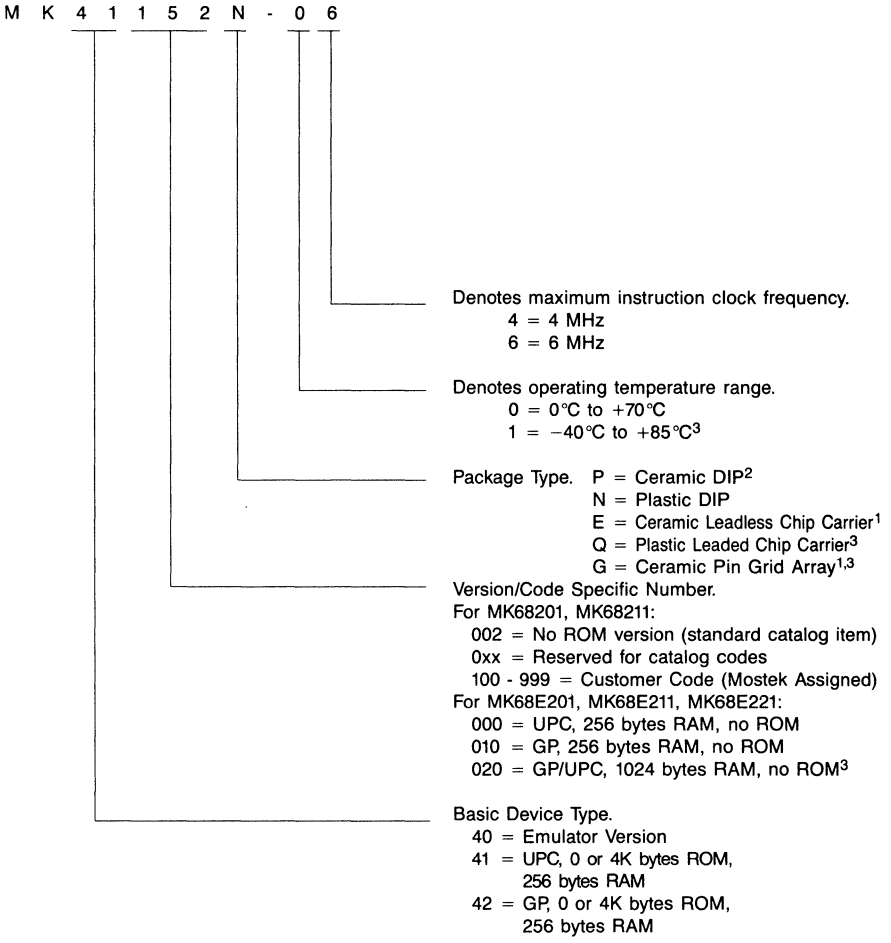


NOTES

1. Available for emulator only.
2. Must be "0" when specifying the emulator.
3. Contact Mostek for availability.

Device Order Number

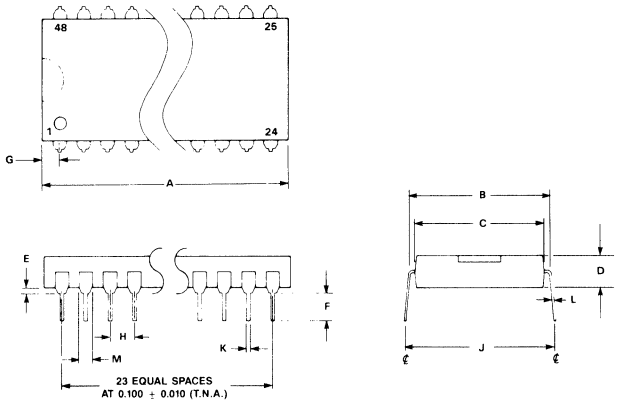
An example of the device order number is shown below:



NOTES
1. Available for emulator only.
2. Intended for prototype orders only.
3. Contact Mostek for availability.

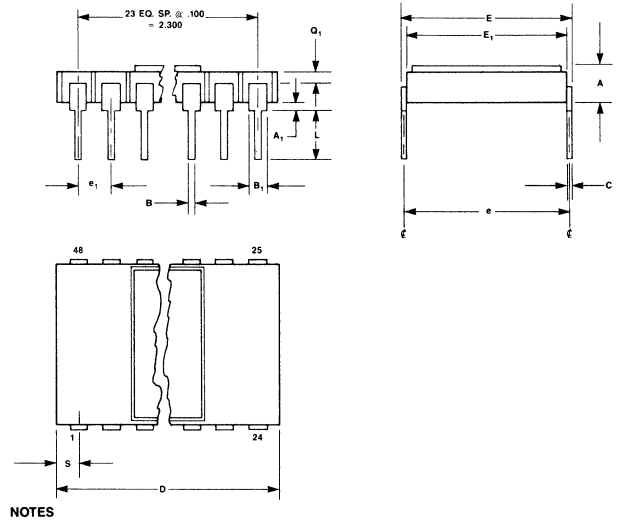
PART NUMBER EXAMPLES (A noninclusive list)	
MK68201/44N-04	Device Order Number = MK41XXXN-04 Speed = 4MHz Temperature = 0° to 70°C Package = 48 pin plastic RAM = 256 bytes ROM = 4096 bytes Bus = UPC
MK68211/04N-06	Device Order Number = MK42002N-06 Speed = 6MHz Temperature = 0° to 70°C Package = 48 pin plastic RAM = 256 bytes ROM = None Bus = GP
MK68E211/04E-14	Device Order Number = MK40010E-14 Speed = 4MHz Temperature = −40° to +85°C Package = 84 pin ceramic LCC RAM = 256 bytes ROM = None Bus = GP
MK68E221/0CG-06	Device Order Number = MK40020G-06 Speed = 6MHz Temperature = 0° to 70°C Package = 84 lead PGA RAM = 1024 bytes ROM = None Bus = GP/UPC

MK68200 48-Pin Plastic Dual-In-Line Package (N)



	MILLIMETERS		INCHES	
DIM.	MIN	MAX	MIN	MAX
A	61.468	62.738	2.420	2.470
B	14.986	16.256	0.590	0.640
C	13.462	13.97	0.530	0.550
D	3.556	4064	0.140	0.160
E	0.381	1.524	0.015	0.060
F	3048	3.81	0.120	0.150
G	1.524	2.286	0.060	0.090
H	1.186	1.794	0.090	0.110
J	15.24	17.78	0.600	0.700
K	0.381	0.533	0.015	0.021
L	0.203	0.305	0.008	0.012
M	1.143	1.778	0.045	0.070

MK68200 48-Pin Ceramic Dual-In-Line Package (P)

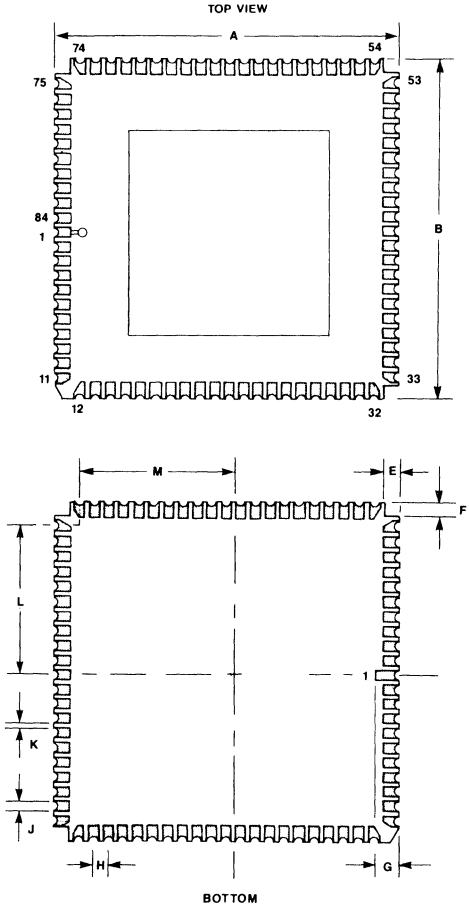


	INCHES		
DIM.	MIN.	MAX.	NOTES
A	.085	.190	
A_1	.020	.070	1
B	.015	.023	
B_1	.038	.060	
C	.008	.012	
D	2.370	2.430	
E	.595	.625	
E_1	.580	.610	
e	.590	.700	2
e_1	.100		
L	.120	.170	
Q_1	.010		
S	.035	065	

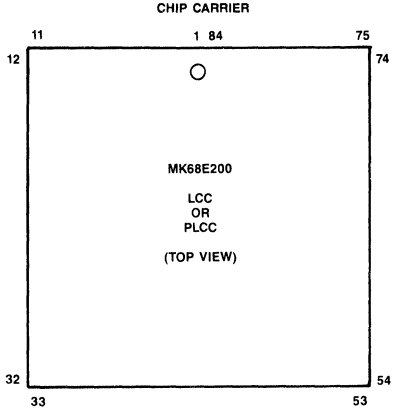
NOTES

1. Package stand off to be measured per JEDEC requirements.
2. Measured from centerline to centerline at lead tips.

MK68E200 84-Pin
Ceramic Leadless Chip Carrier (E)



DIM.	INCHES	
	MIN	MAX
A	1.138	1.167
B	1.138	1.167
C	0.070	0.090
D	0.080	0.110
E	0.044	0.056
F	0.044	0.056
G	0.075	0.095
H	0.048	0.052
J	0.033	0.039
K	0.010	0.018
L	0.495	0.505
M	0.495	0.505



LCC	FUNCTION	LCC	FUNCTION	LCC	FUNCTION	LCC	FUNCTION
1	P1-4/XIO	24	P4-8/RDYL	44	P0-6	65	PB-9
2	P1-5/XI1	25	P4-9/RDYH	45	P0-5	66	PB-10
3	P1-6/XI2	26	P4-10/STRL	46	P0-4	67	PB-11
4	P1-7	27	P4-11/STRH	47	P0-3	68	PB-12
5	P1-8	28	MODE	48	P0-2	69	PB-13
6	P1-9	29	CLK2	49	P0-1	70	PB-14
7	P1-10	30	CLK2	50	P0-0	71	PB-15
8	P1-11	31	CLKOUT	51	FC2 +	72	P4-13/TAI
13	GP/$\overline{\text{UPC}}$ +	32	FC1 +	54	VCC	73	P4-12/TBI
14	P1-12	34	GROUND	55	GROUND	76	P4-15/TAO
15	P1-13	35	P0-15	56	PB-0	77	P4-14/TBO
16	P1-14	36	P0-14	57	PB-1	78	$\overline{\text{RESET}}$
17	P1-15	37	P0-13	58	PB-2	79	$\overline{\text{NMI}}$
18	$\overline{\text{PBLB}}$	38	P0-12	59	PB-3	80	P1-0/SO
19	$\overline{\text{PBHB}}$	39	P0-11	60	PB-4	81	P1-1/TCLK
20	PBR/$\overline{\text{W}}$	40	P0-10	61	PB-5	82	P1-2/RCLK
21	$\overline{\text{PBDTACK}}$	41	P0-9	62	PB-6	83	P1-3/SI
22	$\overline{\text{PBAS}}$	42	P0-8	63	PB-7	84	VCC
23	$\overline{\text{PBDS}}$	43	P0-7	64	PB-8		

+ AVAILABLE ON MK68E221 (1K RAM VERSION)

Figure 36. MK68E200 Pin Assignment, Emulator Version

THOMSON COMPONENTS
MOSTEK

MK68HC200 16-BIT

SINGLE-CHIP MICROCOMPUTERS

MK68HC201/MK68HC211/MK68HC221

PRELIMINARY

MICROCOMPUTER COMPONENTS

FEATURES

- ☐ 16-bit high performance, single-chip microcomputer
- ☐ Modular architecture
- ☐ Power saving stop and idle modes
- ☐ 14 address and data registers
  - — Eight 16-bit or sixteen 8-bit data registers
  - — Six 16-bit address registers
- ☐ Advanced 16-bit instruction set
  - — Bit, byte and word operands
  - — Nine addressing modes
  - — Byte and word BCD arithmetic
- ☐ High performance (12.5 MHz instruction clock)
  - — 0.24 μs register-to-register move or add
  - — 1.68 μs 16 × 16 multiply
  - — 1.84 μs 32/16 divide
- ☐ Available with 0 or 4k bytes of ROM
- ☐ 256 bytes of RAM
- ☐ Three 16-bit timers
  - — Interval modes
  - — Event modes
  - — One-shot modes
  - — Pulse and period measurement modes
- ☐ Serial channel
  - — Double buffered receive and transmit
  - — Asynchronous to 781Kbps
  - — Synchronous to 3.125Mbps
  - — Address wake-up recognition and generation
  - — Internal/external baud rate generation
- ☐ Parallel I/O
  - — Up to 40 pins
  - — Direction programmable by bit
  - — One 16-bit or 2 8-bit port(s) with handshaking
- ☐ Interrupt controller
  - — 16 independent vectors
  - — Expandable to handle an unlimited number of interrupts
  - — Eight external interrupt sources
  - — One non-maskable interrupt
  - — Individual interrupt masking
- ☐ Optional external bus
  - — 16-bit multiplexed address/data bus
  - — Automatic bus request/grant arbitration
  - — Two control bus versions
    - • 68000 compatible bus (UPC) (MK68HC201)
    - • General Purpose bus (GP) (MK68HC211)
- ☐ Emulator version available
  - — Added private bus
  - — No on-chip ROM
  - — 512 bytes on-chip RAM
  - — GP or UPC bus version with one part (MK68HC221)
- ☐ 16, 20, and 25 MHz time base versions produce 8, 10 and 12.5 MHz instruction clock rates respectively
  - — Crystal or external TTL clock
- ☐ Single +5 volt power supply
- ☐ Plastic or Ceramic Chip carrier packaging

GENERAL DESCRIPTION

MK68HC200 designates a series of new high-performance, 16-bit, single-chip microcomputers from Thomson Components - Mostek Corporation. Implemented in 1.5 micron HCMOS technology, they incorporate an architecture designed for superior performance in computation-intensive control applications. A modern, comprehensive instruction set (which features both high speed execution and code space efficiency) is combined on-chip with extensive, flexible I/O capabilities. On-chip RAM and optional on-chip ROM are provided with a full 64K byte addressing space.

The MK68HC200 can be used to design a true application specific microcontroller. The circuit is partitioned into three major functional blocks: CPU, memory, and I/O. The CPU is the core of the circuit and communicates with the memory via the memory address and data bus, and with the I/O via the I/O bus. New I/O or

memory modules can be designed and added to the CPU core to customize the MK68HC200 for a particular application. The initial product offerings in the MK68HC200 family will contain I/O and memory features listed above. This is consistent with the features available on the NMOS MK68200. Future product offerings will contain various assortments of on-chip I/O and memory modules.

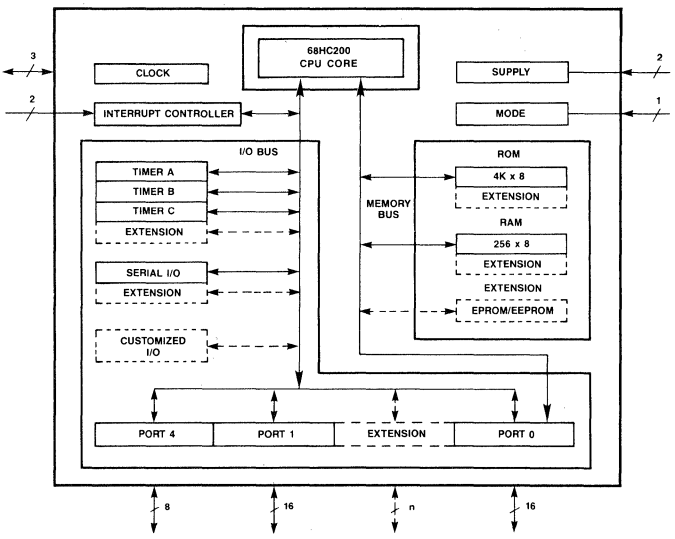


Figure 1. MK68HC200 - Modular Architecture Concept

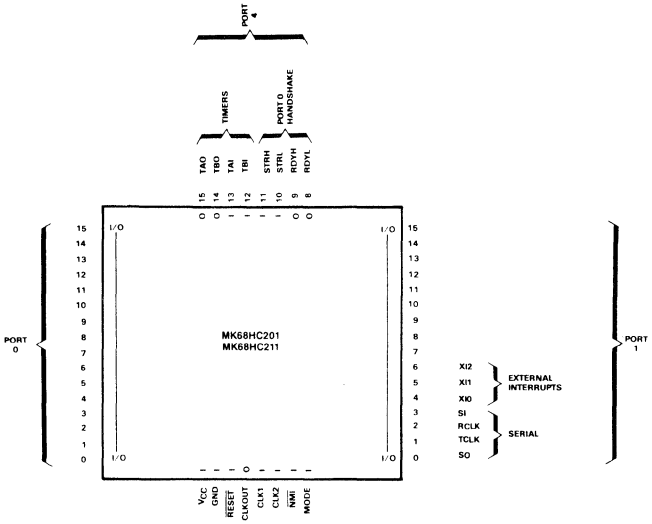


Figure 2. MK68HC201, MK68HC211 Logical Pinout, Single-Chip Mode

SINGLE-CHIP DESCRIPTION

Figure 2 illustrates the functions of specific pins for an MK68HC201 or MK68HC211, operating in a single-chip mode. When the device is operating in one of the expanded bus modes, the pins on Port 0 become the multiplexed address/data bus, and the upper half of Port 1 becomes the control signals (GP or UPC) for the bus. The following description applies to the pins only when the device is used in the non-expanded or single-chip mode. Descriptions of the pin functions for the expanded bus modes are in the Expanded Bus Operation section of this data sheet.

MNEMONIC	PIN FUNCTIONS FOR SINGLE-CHIP OPERATION
V_{CC}	Supply voltage 4.5 to 5.5 V
GND	Ground
$\overline{RESET}$	Reset (input, active low) - $\overline{RESET}$ input overrides ongoing execution (including interrupts) and resets the chip to its initial power-up condition. $\overline{RESET}$ cannot be masked.
CLKOUT	Clock Output (output) - CLKOUT will output the instruction clock rate, which is one-half of the frequency provided on CLK1 and CLK2.
CLK1, CLK2	Time base inputs (inputs) - CLK1 and CLK2 may be connected to a crystal, or CLK1 may be connected to an external TTL-compatible oscillator while CLK2 is left floating.
$\overline{NMI}$	Non-maskable interrupt (input, active low, negative edge triggered) - The $\overline{NMI}$ request line has a higher priority than all of the maskable interrupts. $\overline{NMI}$ is always enabled regardless of the state of the L1E (Level 1 Interrupt Enable) bit in the Status Register.
MODE	Mode (input) - The MODE pin has three states, which select fully expanded external bus, partially expanded external bus, or no expanded bus (single-chip configuration).
P0-0 - P0-15	Port 0 (input/output) - Each bit in Port 0 may be individually programmed for general purpose input or output. Port 0 also has several handshaking modes to allow parallel, asynchronous communication with other devices. The high and low bytes may be programmed individually or jointly to be inputs, outputs, or bidirectional.
P1-0 - P1-15	Port 1 (input/output) - Each of the 16 bits in Port 1 may be individually programmed for input or output. Additionally, the lowest seven bits of Port 1 may be programmed to serve specific alternate functions as shown below.
P1-6/XI2	External Interrupt 2 (input, rising or falling edge triggered) - The programmer may select the rising or falling edge as active for XI2.
P1-5/XI1	External Interrupt 1 (input, falling edge triggered) - The XI1 may be used to interrupt the MK68HC200 on the falling edge of an input pulse.
P1-4/XI0	External Interrupt 0 (input, low level triggered) - The XI0 interrupt input is level triggered (unlike XI1 and XI2). It may be used to produce an internally vectored interrupt or to cause an external fetch of an interrupt vector number when the MK68HC200 is used in an expanded mode with the GP bus.
P1-3/SI	Serial Input (input, active high) - SI is used to receive serial data when the receiver is enabled.
P1-2/RCLK	Receive Clock (input/output, active high) - Depending on the mode programmed, RCLK can be used by the serial port as either an input or an output pin. When used as an input pin, RCLK provides the receive clock and/or the transmit clock. When RCLK is not providing the transmit or receive clock, it can be used as an output for Timer C. In this mode, the receive clock is being provided by Timer C.
P1-1/TCLK	Transmit Clock (input/output, active high) - Depending on the mode programmed, TCLK can be used by the serial port as either an input or an output pin. When used as an input pin, TCLK provides the transmit clock. When TCLK is not providing the transmit clock, it can be used as an output for the Timer C. In this mode, the transmit clock is being provided by either Timer C or RCLK.
P1-0/SO	Serial Output (output, active high) - SO is used to transmit serial data when the transmitter is enabled.

MNEMONIC	PIN FUNCTIONS FOR SINGLE-CHIP OPERATION
P4-8 - P4-15	Port 4 (inputs and outputs) - P4-8, P4-9, P4-14, and P4-15 may be used as general purpose outputs, and P4-10, P4-11, P4-12, and P4-13 may be used as general purpose inputs. Interrupts may be generated on the positive transitions on P4-10 and P4-11. Depending on the mode selected, interrupts may be generated on the positive or the negative transitions on P4-12, and they may be generated on the positive, negative or combined transitions on P4-13. Additionally, these bits may be programmed to serve specific alternate functions, as listed below.
P4-15/TAO	Timer A Output (output) - TAO may be programmed for special functions in the interval, event, and pulse/period modes for Timer A. In the interval mode, TAO's state is determined by the Timer A latch (high and low) that is currently active. That is, if the counter is using the high latch for comparison, TAO is high. In the event mode, TAO is initialized to a "1" state and toggles each time the counter matches the Timer A high latch. In the pulse/period modes, TAO is initialized to a "1" state and toggles on positive transitions on TAI.
P4-14/TBO	Timer B Output (output) - TBO may be programmed for special functions in the interval and one-shot modes for Timer B. In the interval mode, TBO is initialized to a "1" state and toggles each time the counter matches the Timer B latch value. In the one-shot modes TBO is initialized to a "1" state, and the counter begins counting in response to the occurrence of an active edge on TBI. TBO will not go low until the counter matches the value loaded into the Timer B latch.
P4-13/TAI	Timer A Input (input, positive and/or negative edge triggered) - TAI may be programmed for special functions in the event mode or the pulse/period modes for Timer A. In the event mode, the counter is incremented on each active transition (positive or negative edge programmable) on TAI. In the pulse/period modes, the counter measures the time during which the signal on TAI remains high and low.
P4-12/TBI	Timer B Input (input, positive or negative edge triggered) - TBI may be programmed for special functions for the Timer B one-shot modes. In the one-shot modes, TBI acts as a trigger input.
P4-11/STRH, P4-10/STRL	Strobe High Byte, Strobe Low Byte (input, active high) - STRH and STRL are both used for input, output and bidirectional handshaking on Port 0. These signals are issued by the peripheral to acknowledge the receipt of data made available by the MK68HC200, or are issued by the peripheral to load data from the peripheral into the Port 0 input register.
P4-9/RDYH, P4-8/RDYL	Ready High Byte, Ready Low Byte (output, active high) - RDYH and RDYL are used for input, output, and bidirectional handshaking on Port 0. The ready signal goes active to indicate that peripheral data is stable and ready for transfer to the peripheral or is used when Port 0 is empty and is ready to accept data from the peripheral.

MK68HC200 APPLICATIONS

The MK68HC200 is designed to serve the needs of a wide variety of control applications, which require high performance operation with a minimal parts count implementation. Industrial controls, instrumentation, and intelligent computer peripheral controls are all examples of applications served by the MK68HC200. High speed mathematical ability, rapid I/O addressing and interrupt response, and powerful bit manipulation instructions provide the necessary tools for these applications. In addition to its single-chip microcomputer configuration, both distributed intelligence and parallel multiprocessing system configurations are supported by the MK68HC200, as illustrated in Figures 10 and 11.

In applications requiring loosely-coupled distributed intelligence, several MK68HC200's may be interconnected on a common serial network. The on-chip USART supports a wake-up mode in which an additional bit is appended to the data stream to distinguish a serial data word as address or data. The wake-up logic prevents the serial channel from generating interrupts unless certain criteria have been met. The wake-up options available are: Wake-up on any address or data character, wake-up on any address, or wake-up on address match.

Alternately, the MK68HC200 may be configured as an expandable CPU device which can access external memory and I/O resources. In this operating mode, parallel I/O pins are replaced by multiplexed address/data and control lines. Bus arbitration logic is incorporated on the chip to support a direct interface in parallel shared bus multiprocessor system configurations. Two versions exist which support two types of con-

trol signals present on the expanded bus configuration. The General Purpose (GP) bus option allows the MK68HC200 to operate either as an executive or a peripheral processor. As an executive processor, the MK68HC200 can control an external system bus and grant the use of it to requesting devices, such as DMA controllers and/or peripheral processors. As a peripheral control processor, the MK68HC200 can provide intelligent local control of an I/O device in a computer system and, thereby, relieve the executive processor of these tasks. In this configuration, the MK68HC200 has the capability of effectively performing DMA transfers between system memory and the I/O device. The on-chip resources of ROM, RAM, and I/O are accessed within the MK68HC200 without affecting utilization of the shared system bus. Therefore, only external communications compete for bus bandwidth.

The Universal Peripheral Controller (UPC) bus option supports a direct interface to a 68000 executive processor. Thus, the MK68HC200 can be used as a cost-effective intelligent peripheral controller in 68000 systems. The UPC version's direct bus interface to the 68000 makes the MK68HC200 particularly well-suited for performing many intelligent I/O functions in a 68000 system. For example, since the MK68HC200 includes both a serial channel and an external bus capable of performing DMA transfers, it can be programmed to act as serial protocol controller with DMA capability.

For additional information on the MK68HC200 refer to the MK68HC200 Principles of Operation Manual, publication number 4430196.

PROCESSOR ARCHITECTURE

The MK68HC200 microcomputer contains an advanced processor architecture, combining the best properties of both 8- and 16-bit processors. A large majority of instructions operate on either byte or word operands. Figure 3 summarizes the internal architecture of the MK68HC201 and MK68HC211.

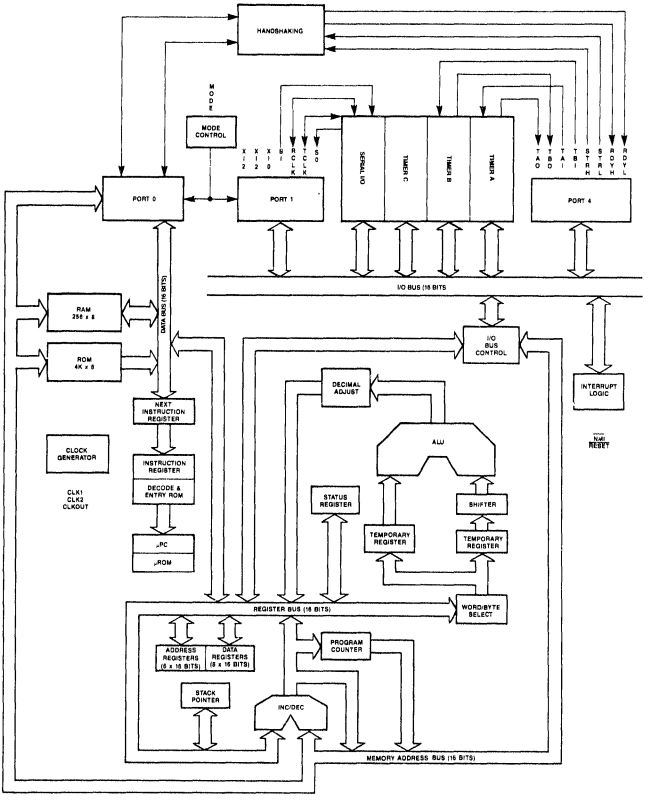


Figure 3. MK68HC201, MK68HC211 Block Diagram

REGISTERS

The MK68HC200 register set includes three system registers, six address registers, and eight data registers. The three 16-bit system registers (Figure 4) include a Program Counter, a Status Register, and a Stack Pointer. The six address registers may be used either for 16-bit data or for memory addressing. The eight 16-bit data registers are used for data and may be referenced as sixteen 8-bit registers, providing great flexibility in register allocation.

ADDRESSING

The MK68HC200 directly addresses a 64K byte memory space, which is organized as 32K 16-bit words. The memory is byte-addressable, but most transfers occur 16 bits at a time, for increased performance over 8-bit microcomputers. All input/output is memory-mapped, and the on-chip I/O is situated in the top 1K bytes of the address space. In the single-chip mode, all resources including ROM, RAM, and I/O, are accessed via an internal or private bus. The memory map, which is accessed by this bus in the single-chip mode, is depicted in Figure 5. Note on-chip RAM always begins at $FBFF and extends downward. ROM always begins at zero and extends upward.

Nine addressing modes provide ease of access to data in the MK68HC200, as depicted in Table 1. The four register indirect forms utilize the address registers and the Stack Pointer and support many common data structures such as arrays, stacks, queues, and linked lists. I/O Port addressing is a short form addressing mode for the first 16 words of the I/O port space and allows most instructions to access the most often referenced I/O ports in just one word. Many microcomputer applications are I/O intensive and short, fast addressing of I/O has a significant impact on performance.

INSTRUCTION SET

The MK68HC200 instruction set has been designed with regularity and ease of programming in mind. In addition, instructions have been encoded to minimize code space, a feature which is especially important in single-chip microcomputers. Small code space is related to execution speed, and most instructions execute in either three or six instruction clock periods. See Table 2.

In addition to operations on bytes and words, the MK68HC200 has rapid bit manipulation instructions that can operate on registers, memory, and ports. The bit to be affected may be an immediate operand of the instruction, or it may be dynamically specified in a register. Operations available include bit set, clear, test, change, and exchange; and all bit operations perform a bit test as well. Since each instruction is indivisible, this provides the necessary test-and-set function for the implementation of semaphores.

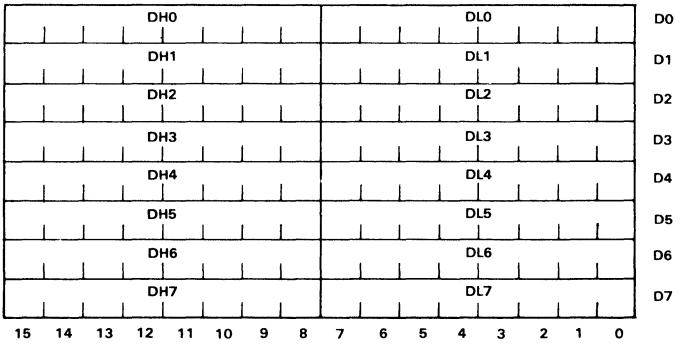
The MOVE group of instructions has the most extensive capabilities. A wide variety of addressing mode combinations is supported including memory-to-memory transfers. A special move multiple is included to save and restore a specified portion of the registers rapidly.

In total, the MK68HC200 instruction set provides a programming environment, similar to the 68000, which has been optimized for the needs of the single-chip microcomputer marketplace. A summary of the instruction set is provided in Table 3.

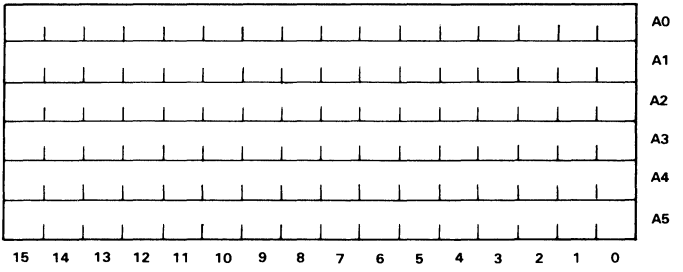
Table 1. Addressing Modes

Register
Register Indirect
Register Indirect with Post-increment
Register Indirect with Pre-decrement
Register Indirect with Displacement
Program Counter Relative
Memory Absolute
Immediate
I/O Port

DATA REGISTERS:



ADDRESS REGISTERS:



SYSTEM REGISTERS:

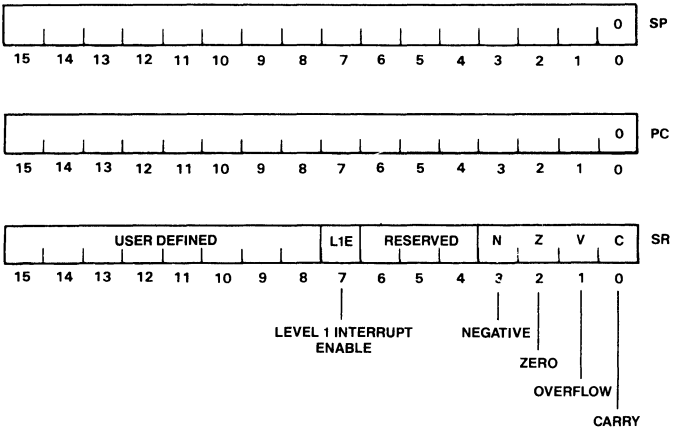


Figure 4. Register Set

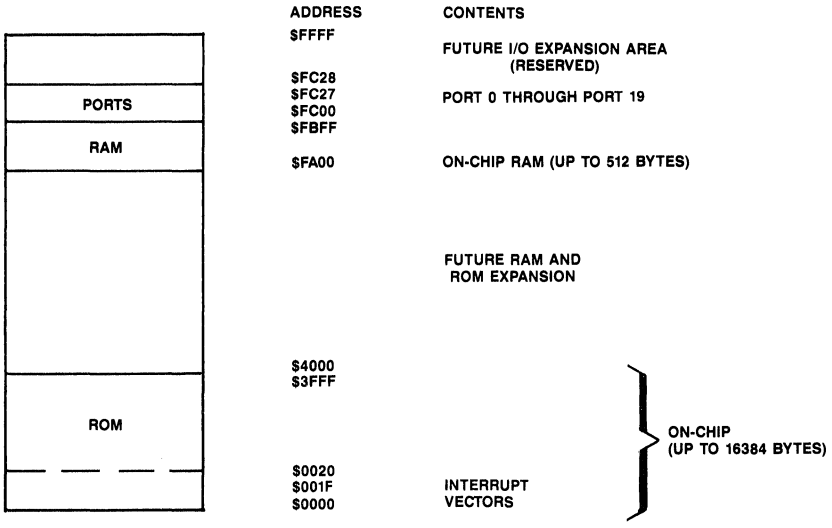


Figure 5. Addressing Space For Single-Chip Configuration

Table 2. Instruction Execution Times

Instruction Type	Clock Periods	Execution Time with 8 MHz Clock (μs)	Execution Time with 10 MHz Clock (μs)	Execution Time with 12.5 MHz Clock (μs)
Move Register-to-register	3	0.38	0.30	0.24
Add Register-to-register (binary or BCD)	3	0.38	0.30	0.24
Move Memory-to-register	6	0.75	0.60	0.48
Add Register-to-memory	9	1.13	0.90	0.72
Multiply (16 × 16)	21	2.63	2.10	1.68
Divide (32/16)	23	2.88	2.30	1.84
Move Multiple (save or restore all registers)	55	6.88	5.50	4.40

Table 3. Instruction Set Summary

INSTRUCTION	DESCRIPTION	INSTRUCTION	DESCRIPTION
ADD	ADD	JMPA	JUMP ABSOLUTE
ADD.B	ADD BYTE	JUMPR	JUMP RELATIVE
ADDC	ADD WITH CARRY	LIBA	LOAD INDEXED BYTE ADDRESS
ADDC.B	ADD WITH CARRY BYTE	LINK	LINK
AND	LOGICAL AND	LIWA	LOAD INDEXED WORD ADDRESS
AND.B	LOGICAL AND BYTE	LSR	LOGICAL SHIFT RIGHT
ASL	ARITHMETIC SHIFT LEFT	LSR.B	LOGICAL SHIFT RIGHT BYTE
ASL.B	ARITHMETIC SHIFT LEFT BYTE	MOVE	MOVE
ASR	ARITHMETIC SHIFT RIGHT	MOVE.B	MOVE BYTE
ASR.B	ARITHMETIC SHIFT RIGHT BYTE	MOVEM	MOVE MULTIPLE REGISTERS
BCHG	BIT CHANGE	MOVEM.B	MOVE MULTIPLE REGISTERS BYTE
BCLR	BIT CLEAR	MULS	MULTIPLY SIGNED
BEXG	BIT EXCHANGE	MULU	MULTIPLY UNSIGNED
BSET	BIT SET	NEG	NEGATE
BTST	BIT TEST	NEG.B	NEGATE BYTE
CALLA	CALL ABSOLUTE	NEGC	NEGATE WITH CARRY
CALLR	CALL RELATIVE	NEGC.B	NEGATE WITH CARRY BYTE
CLR	CLEAR	NOP	NO OPERATION
CLR.B	CLEAR BYTE	NOT	ONE'S COMPLEMENT
CMP	COMPARE	NOT.B	ONE'S COMPLEMENT BYTE
CMP.B	COMPARE BYTE	OR	LOGICAL OR
DADD	DECIMAL ADD	OR.B	LOGICAL OR BYTE
DADD.B	DECIMAL ADD BYTE	POP	POP
DADDC	DECIMAL ADD WITH CARRY	POPM	POP MULTIPLE REGISTERS
DADDC.B	DECIMAL ADD WITH CARRY BYTE	PUSH	PUSH
DI	DISABLE INTERRUPTS	PUSHM	PUSH MULTIPLE REGISTERS
DIVU	DIVIDE UNSIGNED	RET	RETURN FROM SUBROUTINE
DJNZ	DECREMENT COUNT AND JUMP IF NON-ZERO	RETI	RETURN FROM INTERRUPT
		ROL	ROTATE LEFT
DJNZ.B	DECREMENT COUNT BYTE AND JUMP IF NON-ZERO	ROL.B	ROTATE LEFT BYTE
		ROLC	ROTATE LEFT THROUGH CARRY
DNEG	DECIMAL NEGATE	ROLC.B	ROTATE LEFT THROUGH CARRY BYTE
DNEG.B	DECIMAL NEGATE BYTE		
DNEGC	DECIMAL NEGATE WITH CARRY	ROR	ROTATE BYTE
DNEGC.B	DECIMAL NEGATE WITH CARRY BYTE	ROR.B	ROTATE RIGHT BYTE
		RORC	ROTATE RIGHT THROUGH CARRY
DSUB	DECIMAL SUBTRACT	RORC.B	ROTATE RIGHT THROUGH CARRY BYTE
DSUB.B	DECIMAL SUBTRACT BYTE		
DSUBC	DECIMAL SUBTRACT WITH CARRY	STOP	STOP
DSUBC.B	DECIMAL SUBTRACT WITH CARRY BYTE	SUB	SUBTRACT
		SUB.B	SUBTRACT BYTE
EI	ENABLE INTERRUPTS	SUBC	SUBTRACT WITH CARRY
EOR	EXCLUSIVE OR	SUBC.B	SUBTRACT WITH CARRY BYTE
EOR.B	EXCLUSIVE OR BYTE	TEST	TEST
EXG	EXCHANGE	TEST.B	TEST BYTE
EXG.B	EXCHANGE BYTE	TESTN	TEST NOT
EXT	EXTEND SIGN	TESTN.B	TEST NOT BYTE
HALT	HALT	UNLK	UNLINK
IDLE	IDLE		

INPUT/OUTPUT ARCHITECTURE

The I/O capabilities of the MK68HC200 are extensive, encompassing timers, a serial channel, parallel I/O, and an interrupt controller. All of these devices are accessible to the programmer as ports within the top 1K bytes of the address space, and the most commonly accessed ports may be accessed with the short port addressing mode. A description of these ports is given in Table 4.

In total, 40 pins of the 48 are used for I/O, and their functions are highly programmable by the user. In particular, many pins can perform multiple functions, and the programmer selects which ones are to be used. For example, TAI may be used as an input for Timer A, an interrupt source, or a general purpose input pin. The interrupt source may be selected simultaneously with either of the other functions.

Table 4. Port Descriptions

PORT	ADDRESS	READ/WRITE	BYTE-ADDRESSABLE	FUNCTION
0	$FC00	READ/WRITE	YES	16 EXTERNAL I/O PINS OR ADDRESS/DATA BUS
1	$FC02	READ/WRITE	YES	16 EXTERNAL I/O PINS (INCLUDING INTERRUPT, SERIAL I/O PINS, AND BUS CONTROL)
2	$FC04	—	—	(RESERVED)
3	$FC06	LOW BYTE: READ/WRITE HIGH BYTE: READ	YES	SERIAL TRANSMIT (LOW BYTE) AND RECEIVE (HIGH BYTE) BUFFER
4	$FC08	INPUTS: READ ONLY OUTPUTS: READ/WRITE	NO	8 EXTERNAL I/O PINS (TIMER CONTROL AND PORT 0 HANDSHAKE CONTROL)
5	$FC0A	—	—	(RESERVED)
6	$FC0C	—	—	(RESERVED)
7	$FC0E	READ/WRITE	NO	INTERRUPT LATCH REGISTER
8	$FC10	READ/WRITE	NO	INTERRUPT MASK REGISTER
9	$FC12	STATUS: READ ONLY CONTROL: READ/WRITE	NO	SERIAL I/O RECEIVE CONTROL AND STATUS
10	$FC14	STATUS: READ ONLY CONTROL: READ/WRITE	NO	SERIAL I/O TRANSMIT CONTROL AND STATUS
11	$FC16	READ GETS COUNTER WRITE GOES TO LATCH	NO	TIMER B LATCH
12	$FC18	READ GETS COUNTER OR LATCH WRITE GOES TO LATCH	NO	TIMER A, LOW LATCH
13	$FC1A	READ GETS COUNTER OR LATCH WRITE GOES TO LATCH	NO	TIMER A, HIGH LATCH
14	$FC1C	READ/WRITE	NO	TIMER AND HANDSHAKE CONTROL
15	$FC1E	STATUS: READ ONLY CONTROL: READ/WRITE	NO	EXPANDED BUS CONTROL AND STATUS
16	$FC20	READ/WRITE	NO	PORT 0 DIRECTION CONTROL (DDR0)
17	$FC22	READ/WRITE	NO	PORT 1 DIRECTION CONTROL (DDR1)
18	$FC24	READ/WRITE	NO	SERIAL I/O MODE AND SYNC REGISTER
19	$FC26	READ GETS COUNTER WRITE GOES TO LATCH AND COUNTER	NO	TIMER C LATCH

TIMERS

The MK68HC200 includes three on-chip timers, each with unique features. They are denoted Timer A, Timer B, and Timer C. All three timers are a full 16 bits in width, and count at the instruction clock rate of the MK68HC200 processor. Thus, this rate provides a resolution equal to the instruction clock period (tc) of the MK68HC200. The maximum count interval is equal to tc $*2^{16}$. Each timer has the capability to interrupt the processor when it matches a predetermined value stored in an associated latch.

Timer A is capable of operating in interval, event, or two pulse/period modes. There is one 16-bit counter and two 16-bit latches, a high latch (Port 13), and a low latch (Port 12), associated with Timer A. Once Timer A is initialized in the interval mode, the counter is reset, then increments at the instruction clock rate until the value loaded into the high latch is reached. The counter is then reset, increments until the low latch value is reached, and the cycle is repeated. In the event mode, the counter is incremented for every active edge on TAI (programmable as positive or negative) until the value in the high latch is reached. The counter is then reset, and the cycle repeats. In the pulse/period modes, the times are measured during which the applied pulse stays high and low. The counter is reset on the occurrence of any transition on TAI, and increments at the instruction clock rate until the occurrence of the next transition. The value in the counter at the end of the high level or low level time is loaded into the appropriate latch. Interrupts may be generated each time the counter reaches the high latch or low latch value in the interval mode or when the counter reaches the high latch in the event mode. Also, an interrupt is generated whenever the counter overflows. See the Pin Description section of the data sheet for TAI and TAO functions in the various Timer A modes.

Timer B is capable of operating in interval or one-shot modes. There is one 16-bit counter and one 16-bit latch (Port 11) associated with Timer B. In the interval mode, the counter is initially reset and incremented at the instruction clock rate until the value in the latch is reached. The counter is then reset, and the cycle repeats. In the one-shot modes, the counter begins incrementing in response to an active transition (programmable as positive or negative) on TBI. The counter is reset when the value in the Timer B latch is reached. In the retriggerable one-shot mode, active transitions on TBI always cause the counter to reset and begin incrementing. In the non-retriggerable one-shot mode, active transitions on TBI have no effect until the counter reaches the latch value. Interrupts may be generated each time the counter reaches the latch value. See the Pin Description section of this data sheet for TBI and TBO functions in the various Timer B modes.

Timer C has a 16-bit down counter and latch (Port 19) associated with it and operates only in the interval mode. The output of Timer C toggles each time the counter value rolls over from 0 to the latch value and may be used to internally supply the baud rate clock for the serial port. Also, an interrupt may be generated each time the counter rolls over to the latch value. Timer C may be output on the TCLK pin (P1-3), depending on the mode programmed.

A detailed description of the Timer Control Port is given on the next page.

Table 5. Timer Modes

Timer	Modes
A	Interval
A	Event
A	Pulse Width and Period Measurement
B	Interval
B	Retriggerable One-shot
B	Non-retriggerable One-shot
C	Interval
C	Baud Rate Generation

PORT 14—Timer Control Register; read/write
$FC1C

HANDSHAKE CONTROL	TCE	TCOC	TAM1	TAM0	TAE	TAIC	TAOC	TBM1	TBM0	TBE	TBIC
15 14 13 12 11	10	9	8	7	6	5	4	3	2	1	0

TCE
(Timer C Enable control)

0 = Disables Timer C; all operations are inhibited, and the timer counter is initialized.
1 = Enables Timer C; the timer begins operation as defined by the other Timer C control bits.

TCOC
(Timer C Output control)

0 = When TCO (Port 18) = 1, TCLK is selected for use as a general purpose I/O pin.
1 = When TCO (Port 18) = 1, TCLK is selected for use as an output for Timer C.

TCOC has no effect when TCO (Port 18) = 0.

TAM1, TAM0
(Timer A Mode control)

These bits select the operating mode of Timer A as follows.

TAM1	TAM0	MODE
0	0	Interval
0	1	Event
1	0	Pulse/period 1
1	1	Pulse/period 2

TAE
(Timer A Enable control)

0 = Disables Timer A; all Timer A operations are inhibited, and the timer counter is initialized.
1 = Enables Timer A; the timer begins operation as defined by the other Timer A control bits.

TAIC
(Timer A Input control)

0 = Selects a negative transition as the active edge for TAI.
1 = Selects a positive transition as the active edge for TAI.

TAOC
(Timer A Output control)

0 = Selects TAO as a general purpose output pin.
1 = Selects TAO as an ouput pin associated with Timer A; TAO is initialized low when TAOC is a one and TAE is zero.

TBM1, TBM0
(Timer B Mode control)

These bits select the operating mode of Timer B as described below.

TBM1	TBM0	MODE
0	0	Interval 0 (TBO is not used)
0	1	Interval 1 (TBO is used)
1	0	Retriggerable one-shot
1	1	Non-retriggerable one-shot

TBE (Timer B Enable control)	0= Disables Timer B; all operations are inhibited, and the timer counter is initialized. 1= Enables Timer B; the timer begins operation as defined by the other Timer B control bits.
TBIC (Timer B Input control)	0= Selects a negative transition as active on TBI. 1= Selects a positive transition as active on TBI.

INTERRUPT CONTROLLER

The MK68HC200 interrupt controller provides rapid service of up to 15 interrupt sources, each with a unique internal vector. The lowest 16 words of the address space contain the starting addresses of the service routines of each potential interrupt source and reset, as shown in Figure 6.

Interrupt sources and RESET are prioritized in the order shown in Figure 6, with RESET having the highest priority. When an interrupt is pending it sets the corresponding bit in the interrupt latch located in Port 7. NMI is the only non-maskable interrupt. All of the other sources share an interrupt enable bit in the processor Status Register. This bit is automatically cleared whenever an interrupt is acknowledged. Also, each of these sources has a corresponding individual mask bit located in Port 8. This feature allows selective masking of particular interrupts, including the ability to choose (with minimal software overhead) any priority scheme desired. In fact, 15 levels of nested priority may be programmed.

Note that the XI2 interrupt is detected on either a rising or falling edge, depending upon the status of the XI2C bit (bit 12 in Port 14). An interrupt will be generated on the falling edge if this bit is set to a "0"; however, if the bit is set to a "1", an interrupt will be generated on the rising edge.

PORT 7 —Interrupt Latch Register; read/write
$FC0E

res	NMI	SPARE	XI2I	STRLI	TAOI	TAII	STRHI	RSCI	RNI	XI1I	TBOI	TBII	XI0I	XMTI	TCI
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PORT 8 —Interrupt Mask Register; read/write
$FC10

(res)		SPARE	XI2M	STRLM	TAOM	TAIM	STRHM	RSCM	RNM	XI1M	TBOM	TBIM	XI0M	XMTM	TCM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

VECTOR NUMBER	NAME	MNEMONIC	VECTOR LOCATION	
0	RESET	RESET	0000	
1	NON-MASKABLE INTERRUPT	NMI	0002	LEVEL 2
2	SPARE	SPARE	0004	
3	EXTERNAL INTERRUPT 2	XI2	0006	
4	STROBE LOW	STRL	0008	
5	TIMER A OUTPUT	TAO	000A	
6	TIMER A INPUT	TAI	000C	
7	STROBE HIGH	STRH	000E	
8	RECEIVE SPECIAL CONDITION	RSC	0010	
9	RECEIVE NORMAL	RN	0012	LEVEL 1
A	EXTERNAL INTERRUPT 1	XI1	0014	
B	TIMER B OUTPUT	TBO	0016	
C	TIMER B INPUT	TBI	0018	
D	EXTERNAL INTERRUPT 0	XI0	001A	
E	TRANSMIT	XMT	001C	
F	TIMER C	TC	001E	

Figure 6. Interrupt and Reset Vectors

SERIAL CHANNEL

The serial channel on the MK68HC200 (Figure 7) is a full-duplex USART with double buffering on both transmit and receive. Port 3 High Byte is the Receive Buffer, and Port 3 Low Byte is the Transmit Buffer.

Word length, parity, stop bits, and modes are fully programmable. The asynchronous mode supports bit rates up to 781 Kbps with an external clock and up to 390Kbps with an internal clock. The byte synchronous mode operates up to 3.125Mbps with either an internal or an external clock.

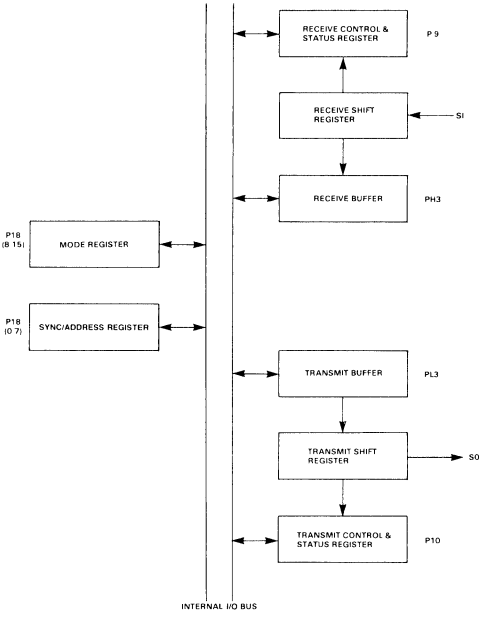


Figure 7. Serial Channel

In addition to the typical USART functions, the serial channel can operate in a special wake-up mode with a wake-up bit appended to each data word, as illustrated in Figure 8. This wake-up bit is used to differentiate normal data words and special address words. The receiver can be programmed to receive only address words or only address words with a specific data value. In this way, the processor can be interrupted only when it receives its particular address and can then change mode to receive the following data words. Wake-up capability is especially useful when several MK68HC200 microcomputers are interconnected on one serial link.

A detailed description of the serial channel control ports is given on the following pages.

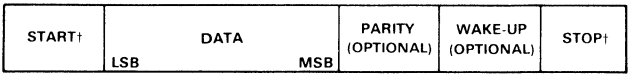


Figure 8. Serial Frame Format

PORT 9 —Serial I/O Receive Control and Status Register;
$FC12 High byte: control register; read/write
Low byte: status register; read only

RE	IS	RW1	RW0	RC	SIS	(res)		BF	OE	PE	FE	SF/AF	(res)		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit Descriptions:

RE
(Receiver Enable control)

0 = Disabled; all status flags cleared.
1 = Enabled.

IS
(Ignore Syncs control)

0 = Disabled; interrupts may occur on all characters received.
1 = Enabled; interrupts cannot occur on sync characters received after the sync match is found.

RW1, RW0
(Receiver Wake-up control)

The receiver wake-up control bits operate as follows.

MODE	RW1	RW0	APPENDED WAKE-UP	BUFFER LOADED	INTERRUPT GENERATED
No Wake up	0	0	no	any character	RN
Wake-up on Any Character	0	1	yes	any character	RN
Wake-up on Address Match	1	0	yes	address match	RSC
Wake-up on Any Address	1	1	yes	any address	RSC

RC
(Receive Clock control)

0 = Selects external receive clock applied on RCLK.
1 = Selects internal clock from the on-chip baud rate generator (Timer C) for the receive clock.

This bit is ignored when either the TCO bit or the LM (Loopback Mode) bit is set.

SIS
(Single Interrupt Select control)

0 = Separate vectors are generated for the Receive Normal and the Receive Special Condition interrupts.
1 = The Receive Normal vector is generated for all receive character interrupts.

BF
(Buffer Full status)

0 = Receive data buffer empty; cleared when receive buffer is read.
1 = Receive buffer full; set when an incoming word is loaded into the receive data buffer.

OE
(Overrun Error status)

0 = No overrun error; cleared when the status register is read.
1 = Overrun error; set when a new word has been received and the previous word has not been read from the receive data buffer.

PE
(Parity Error status)

0 = No Parity Error; cleared when the status register is read.
1 = Parity Error; set when a parity error has been detected on an incoming character in the data stream.

FE
(Frame Error status)

0 = No frame error; cleared when the status register is read.
1 = Frame error; set when a word is transferred to the receive data and no stop bit has been recognized.

The FE flag applies to async formats only.

SF/AF
(Sync Found or Address Found status)

This flag is used for both sync character match conditions and address found conditions in some wake-up modes as follows. SS stands for Sync Search mode.

MODES				
SS	**IS**	**RW1**	**RW0**	**CONDITIONS THAT SET SF/AF**
1	x	x	x	Sync Found on any bit boundry
0	1	x	x	unaffected
0	0	0	0	unaffected
0	0	1	1	Any Address
0	0	1	0	Address Match
0	0	0	1	unaffected

PORT 10—Serial I/O Transmit Control and Status Register
$FC14 High byte: control register; read/write
Low byte: status register; read only

TE	AT	LM	TW1	TW0	TC	P/$\overline{S}$	res	BE	UE	END	(reserved)				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit Descriptions:

TE
(Transmitter Enable control)

0 = Disable the transmitter; any word being shifted out will continue until completion.
1 = Enable the transmitter.

AT
(Automatic Turn Around control)

0 = No effect on TE or RE.
1 = Causes RE to be set to a "1" and TE to be set to a "0" automatically at the end of a transmission.

LM
(Loopback Mode control)

0 = Disables loopback mode.
1 = Causes the transmitter output to be internally connected to receiver input. Also causes Timer C to be used for both the transmit and receive clocks regardless of the state of TC, RC, TCO, and TCOC.

TW1, TW0
(Transmit Wake-up control)

These bits provide control for wake-up operation as follows.

TW1	TW0	OPERATION
1	0	Transmit Data
1	1	Transmit Address
0	X	No Wake-up

TC
(Transmit Clock control)

0 = Selects the external clock signal applied on TCLK for the transmit clock.
1 = Selects the internal baud rate generator output (Timer C) for the transmit clock.

The TC bit is ignored if either the TCO bit or the LM bit is set.

P/$\overline{S}$
(Previous/$\overline{Sync}$ control)

0 = Selects continuous transmission of the contents of the sync character register in the synchronous mode when there is no data to transmit.
1 = Selects continuous transmission of the transmit data buffer in synchronous mode when there is no data to transmit.

BE
(Buffer Empty status)

0 = Transmit Buffer is full; reset to this condition after the transmit buffer is reloaded.
1 = Transmit Buffer is empty; set to this condition after the transmit buffer contents are transferred to the output shift register.

UE
(Underrun Error status)

0 = No underrun error; cleared following a read of the transmit buffer.
1 = Underrun error; set only in the synchronous mode when the last word has been shifted out and transmit buffer has not been reloaded.

END
(End of Transmission status)

0 = No end of transmission; cleared by enabling the transmitter.
1 = End of transmission detected; set when the transmitter is disabled and the last character has been shifted out.

PORT 18—Serial I/O Mode and Sync Register; read/write
$FC24

A/$\overline{S}$	WL1	WL0	ST	PAR1	PAR0	TCO	WS	SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit Descriptions:

A/$\overline{S}$
(Asynchronous/$\overline{\text{Synchronous}}$ mode control)

0 = Selects synchronous operation for the serial port; transmit and receive clocks are divided by 1.
1 = Selects asynchronous operation for serial port; transmit and receive clocks are divided by 16.

WL1, WL0
(Word Length control)

These two bits select the length of the data word as follows.

WL1	WL0	Word Length
0	0	8 bits
0	1	7 bits
1	0	6 bits
1	1	5 bits

ST
(Stop Bit control)

This bit is only used in the asynchronous mode. It selects the number of stop bits transmitted.

ST	Number of Stop Bits
0	1
1	2

PAR1, PAR0
(Parity control)

These two bits provide parity control for both the synchronous and asynchronous modes.

PAR1	PAR0	Parity
0	0	no parity
0	1	fixed "0" parity
1	0	odd parity
1	1	even parity

Note that even parity is defined such that the sum of the data and parity bits is even.

TCO
(Timer C Output mode control)

0= Disables Timer C output mode.
1= Enables Timer C output mode; disables Timer C's use as a baud rate generator when LM = 0; causes transmit and receive clocks to be internally connected to RCLK so that TCLK may be used either as general purpose I/O or as an output for Timer C.

WS
(Wake-up Sense control)

The following table lists the effects of the WS bit.

WS	Wake-up bit	Meaning
0	0	Address Word
0	1	Data Word
1	0	Data Word
1	1	Address Word

SYNC7-SYNC0
(Sync character bits)

These eight bits are used to store the sync character or the device address for the wake-up mode.

PARALLEL I/O AND HANDSHAKING

Two 16-bit ports, P0 and P1, may be used for parallel I/O. If individual bits are desired, each of the 32 bits may be separately defined as input or output. This is achieved by setting the corresponding bits in the Data Direction Registers, Port 16 (Data Direction Register for Port 0) and Port 17 (DDR for Port 1). Bits may be grouped to provide the exact data widths desired.

Eight additional I/O bits are provided in Port 4. Bits 15, 14, 9 and 8 are output only, and bits 13, 12, 11 and 10 are input only.

Port 0 has the additional capability of operating under the control of external handshaking signals. Eight-bit or sixteen-bit sections of P0 may be individually controlled as input, output or bidirectional I/O. This is achieved by programming the handshake control bits as detailed below.

PORT 14— Handshake Control Register; read/write
$FC1C

HSM2	HSM1	HSM0	res	HSE	TIMER CONTROL
15	14	13	12	11	10 9 8 7 6 5 4 3 2 1 0

Bit Descriptions:

HSE (Handshake enable control)
0 = Handshaking is disabled.
1 = Handshaking is enabled.

HSM2, HSM1, HSM0 (Handshake Mode control)
The handshake mode bits operate as follows:

HSM2	HSM1	HSM0	HIGH HANDSHAKE	LOW HANDSHAKE
0	0	0	Inactive	PLO or PO output
0	0	1	PHO output	PLO output
0	1	0	Inactive	PLO input
0	1	1	PHO input	PLO input
1	0	0	PHO input	PLO output
1	0	1	Inactive	PO input (word only)
1	1	0	PLO output	PLO input (bidirectional)
1	1	1	PO output	PO input (bidirectional)

Two pairs of Ready and Strobe signals, which are available as programmable options on Port 4, provide the necessary control for handshaking.

P4-9/RDYH, P4-8/RDYL
(Ready High Byte, Ready Low Byte)
Output, active high. RDYH and RHYL are used for input, output, and bidirectional handshaking on Port 0.

1) Output mode: The ready signal goes active to indicate that the Port 0 output register has been loaded, and the peripheral data is stable and ready for transfer to the peripheral device.

2) Input mode: The ready signal is active when the Port 0 input register is empty and is ready to accept data from the peripheral device.

3) Bidirectional mode: The RDYH signal is active when data is available in Port 0 output register for transfer to the peripheral device. In this mode, data is not placed on the Port 0 data bus unless STRH is active. The RDYL signal is active when the Port 0 input register is empty and is ready to accept data from the peripheral device.

P4-11/STRH, P4-10/STRL
(Strobe High Byte, Strobe Low Byte)
Input, active high. STRH and STRL are both used for input, output, and bidirectional handshaking on Port 0.

1) Output Mode; The positive edge of this strobe is issued by the peripheral to acknowledge the receipt of data made available by the MK68HC200.

2) Input mode: The strobe is issued by the peripheral to load data from the peripheral into the Port 0 input register. Data is latched into the MK68HC200 on the negative edge of this signal.

3) Bidirectional mode: When the STRH signal is active, data from the Port 0 output register is gated onto the Port 0 bidirectional data bus.

The negative edge of STRH acknowledges the receipt of the output data. The negative edge of the signal applied to the STRL signal is used to latch input data into Port 0.

EXPANDED BUS OPERATION

When it is necessary to expand beyond the on-chip complement of RAM, ROM, or I/O, or when operation in a parallel multiprocessing system is desired, the MK68HC200 may be placed in an external bus mode. The MODE pin is used to select the expansion capability on power-up and reset to one of the following states:

MODE PIN

V_{CC} - No expansion (single chip mode)
GND - Partial Expansion
CLKOUT - Full Expansion

By programming the appropriate bits in Port 15 (which are described below), the MK68HC200 may be reconfigured dynamically. In an expansion mode Port 0 becomes the 16-bit multiplexed address/data bus and eight bits from Port 1 become control signals which handle data transfer and bus arbitration. Sixteen lines are still available for I/O functions, including eight lines from Port 1 and all eights lines of Port 4. See figure 9 for the expanded bus pinout. The following page describes the functions of the expanded bus pins.

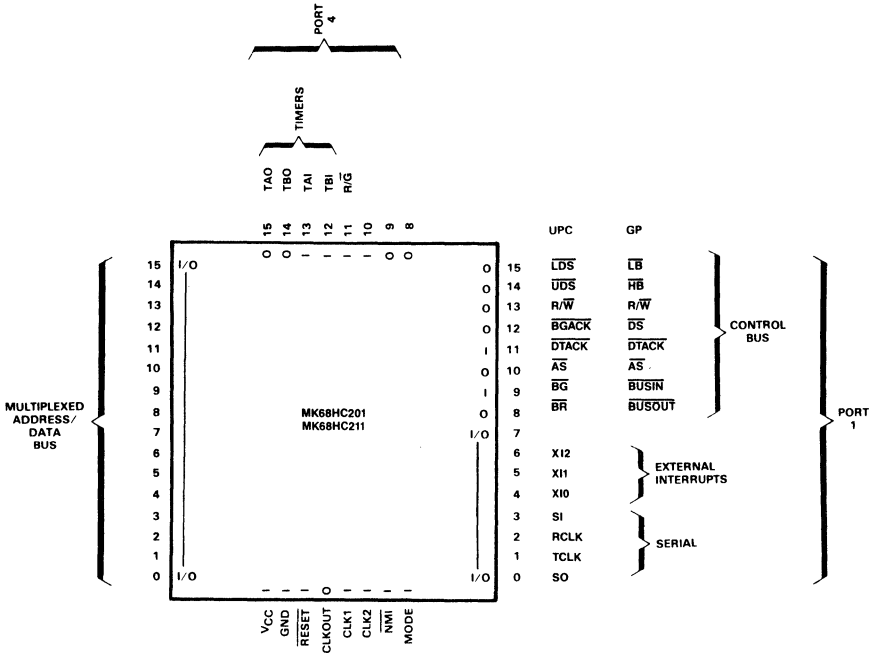


Figure 9. MK68HC201, MK68HC211 Logical Pinout Expanded Bus

MNEMONIC	PIN FUNCTIONS FOR EXPANDED BUS OPERATION (Common for GP and UPC options)
R/$\overline{W}$	Read/Write (output, active high and low) - R/$\overline{W}$ determines whether a read or a write is being performed during the current bus cycle. It is stable for the entire bus operation. A high signal denotes a read, and a low signal denotes a write.
$\overline{DTACK}$	Data Transfer Acknowledge (input, active low) - When the addressed device has either placed the requested read data on the bus or taken the write data from the bus, $\overline{DTACK}$ should be brought low to signify completion. The data portion of the bus cycle will be extended indefinitely until this signal is asserted. For systems using the GP bus, in which no devices need wait states, $\overline{DTACK}$ may be strapped low.
$\overline{AS}$	Address Strobe (output, active low) - $\overline{AS}$ is used to signify that the address is stable on the multiplexed bus. $\overline{AS}$ is high at the beginning of each bus cycle, goes low after the address has stabilized, and returns to the high state near the end of the bus cycle.
MNEMONIC	**PIN FUNCTIONS FOR UPC BUS OPERATION**
$\overline{UDS}$	Upper Data Strobe (output, active low) - $\overline{UDS}$ is used to signify the data portion of the bus cycle for the upper byte of the data bus. For read operations, $\overline{UDS}$ should be used by the external device to gate its most significant byte onto the multiplexed address/data bus. For writes, $\overline{UDS}$ signifies that the upper byte of the bus contains valid data to be written from the processor.
$\overline{LDS}$	Lower Data Strobe (output, active low) - $\overline{LDS}$ is used to signify the data portion of the bus cycle for the lower byte of the data bus. For read operations, $\overline{LDS}$ should be used by the external device to gate its least significant byte onto the multiplexed address/data bus. For writes, $\overline{LDS}$ signifies that the lower byte of the bus contains valid data to be written from the processor.
$\overline{BR}$	Bus Request (output, active low, open drain) - $\overline{BR}$ goes low when the MK68HC200 requires external bus master status.
$\overline{BG}$	Bus Grant (input, active low) - $\overline{BG}$ notifies that the MK68HC200 has been granted the external bus master status.
$\overline{BGACK}$	Bus Grant Acknowledge (output, active low, open drain) - The MK68HC200 will assert $\overline{BGACK}$ when it assumes mastership of the system bus.
MNEMONIC	**PIN FUNCTIONS FOR GP BUS OPERATION**
P4-11/R/$\overline{G}$	Request/$\overline{Grant}$ (input) - During reset, P4-11 served as the R/$\overline{G}$ input (0 = bus grantor, 1 = bus requestor). Following reset, and at all times during program execution, P4-11 may be used as a general purpose input pin.
$\overline{DS}$	Data Strobe (output, active low) - $\overline{DS}$ is used to signify the data portion of the bus cycle. For read operations, $\overline{DS}$ should be used by the external device to gate its contents onto the multiplexed address/data bus. For writes, $\overline{DS}$ signifies that valid data from the processor is on the bus.
$\overline{HB}$	High Byte (output, active low) - $\overline{HB}$ signifies that the upper byte of the data is to be read or written. $\overline{HB}$ remains active for the entire bus cycle.
$\overline{LB}$	Low Byte (output, active low) - $\overline{LB}$ signifies that the lower byte of the data bus is to be read or written. $\overline{LB}$ remains active for the entire bus cycle.
$\overline{BUSIN}$	Bus Input (input, active low) - $\overline{BUSIN}$ provides either bus request or bus grant. When the MK68HC200 is the bus grant device, its $\overline{BUSIN}$ signal is a bus request input from a requesting device on the bus. When the MK68HC200 is a bus request device, its $\overline{BUSIN}$ signal is a bus grant from the granting device on the bus.
$\overline{BUSOUT}$	Bus Output (output, active low) - $\overline{BUSOUT}$ provides the opposite function of $\overline{BUSIN}$. When $\overline{BUSIN}$ is a bus request signal, $\overline{BUSOUT}$ is a corresponding bus grant, and vice versa.

PORT 15— Expanded bus control and status register
$FCIE High byte: read/write
Low byte: read only

SEG1	SEG0	BLCK	F/$\overline{S}$	EXP	F/$\overline{P}$	(res)		UPC/$\overline{GP}$	R/$\overline{G}$	(reserved)					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit Descriptions:

SEG1, SEG0 (Segment bits)	Used in the expanded bus mode when a reference is made to the DMA window. The contents of SEG1 and SEG0 are then output on pins AD15 and AD14, respectively.
BLCK (Bus Lock control)	0 = Disables the bus lock function. 1 = Enables the bus lock function.
F/$\overline{S}$ (Fast/$\overline{\text{Standard}}$ timing control)	0 = Selects standard timing of read/write cycles on the external bus (4 clock periods). 1 = Selects fast timing of read/write cycles on the external bus (3 clock periods.)
EXP (Expanded Mode control)	0 = Expanded mode is disabled. 1 = Expanded mode is enabled.
F/$\overline{P}$ (Full/$\overline{\text{Partial}}$ control)	0 = Partial expand when EXP bit is set to 1. 1 = Full expand when EXP bit is set to 1.
UPC/$\overline{GP}$ (UPC/$\overline{GP}$ status)	0 = Part is programmed in GP mode. 1 = Part is programmed in UPC mode.
R/$\overline{G}$ (Request/$\overline{\text{Grant}}$ status)	0 = Part is programmed bus grantor. 1 = Part is programmed bus requestor.

As shown in figure 9, two different control bus versions are available: a Universal Peripheral Controller (UPC), and which generates 68000 compatible bus signals, and a General Purpose (GP) bus, which can be used to interface to a wide variety of existing microprocessor buses. With the selection of an expanded bus mode, the MK68HC200 can act either as a general purpose CPU chip (bus grant device) or as an intelligent peripheral I/O controller to a host CPU (bus request device). These two system configurations are illustrated in figures 10 and 11.

With the GP bus option, the user may configure the MK68HC200 in either of the two ways shown in figures 10 and 11. As a host CPU (Figure 10), the MK68HC200 bus arbitration logic causes the device to act as the system bus grantor. In other words, the MK68HC200 would have control of the system bus and would grant its use to DMA devices or peripheral CPUs. Alternately the MK68HC200 may be configured as a peripheral CPU (Figure 11) that must issue a request to the bus grant device before being allowed to use the system bus. The selection of one of these two configurations is accomplished by the P4-11 pin at reset time. During reset, P4-11 serves as the R/$\overline{G}$ input (0 = bus grantor, 1 = bus requestor). Following reset and at all times during program execution, P4-11 may be used as a general purpose input pin.

With the GP bus operating in the host CPU configuration, the MK68HC200 may be used to interface with external memory and I/O devices in a manner that is analagous to any general purpose microprocessor. Additionally, the MK68HC200 retains its on chip RAM and I/O resources, with on-chip ROM as an option,

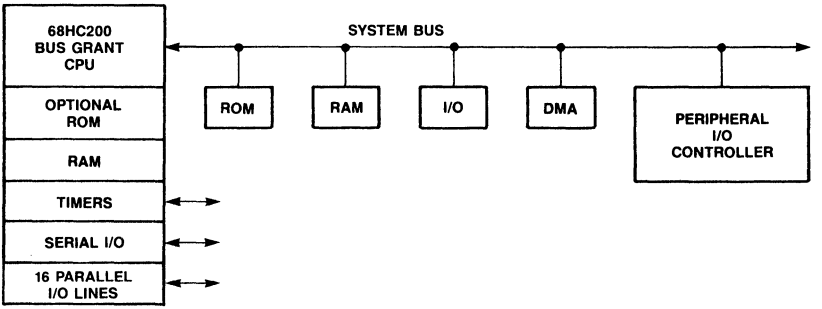


Figure 10. Host CPU Hardware Configuration

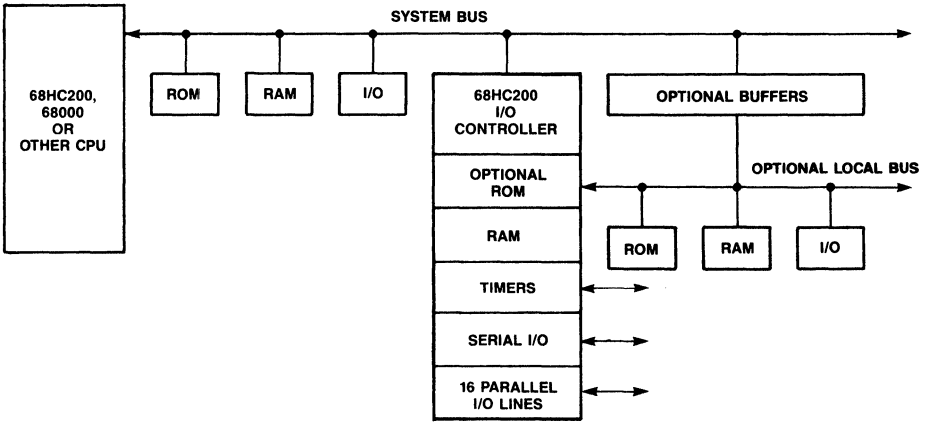


Figure 11. Peripheral I/O Controller Configuration

depending on the expansion configuration selected. $\overline{BUSIN}$ and $\overline{BUSOUT}$ are used to perform the bus arbitration handshake function, where $\overline{BUSIN}$ acts as the bus request input and $\overline{BUSOUT}$ as the bus grant output.

In the full expansion configuration, any on-chip ROM is disabled, and program memory starting at location $0000 is located off-chip and is addressed via the expanded bus, as shown in Figure 13. In effect, the internal bus from locations $0000-$FAFF is mapped onto the external bus. In the partially expanded configuration (Figure 12), on-chip ROM may be accessed on the internal bus. To gain greater addressability in the partial expansion configuration, a scheme is implemented to allow access of a full 64K-byte address space in four segments on the expanded system bus through the 16K byte "window" on the internal bus. Basically, the most significant two bits of address on the expanded bus are replaced with two user-defined segment bits available to the programmer in the expanded bus control and status register, Port 15.

As a peripheral I/O controller, the MK68HC200 operates as a bus requestor that gains mastership of the system bus from the bus grant CPU. The GP bus version may be selected to implement this system configuration in cases where an interface to a general purpose CPU is desired. In this case, the $\overline{BUSIN}$ and $\overline{BUSOUT}$ lines are again used to perform the bus arbitration handshake function, where $\overline{BUSOUT}$ now acts as bus request output, and $\overline{BUSIN}$ acts as bus grant input. In this configuration, the MK68HC200 can conceivably act as a complete peripheral I/O control subsystem on a single chip, with 16 lines of I/O and its on-chip ROM, RAM, timers, and serial I/O performing the necessary interface to the I/O device. The UPC bus version provides the peripheral I/O control function with a direct interface to a 68000 bus grant CPU. Note that the UPC bus version can operate only as a bus request device. Once the MK68HC200 has gained mastership of the system bus via the 68000 bus arbitration handshake lines ($\overline{BR}$, $\overline{BG}$, and $\overline{BGACK}$), it may proceed to perform DMA transfers and communicate with system memory or other I/O devices in the system.

As in the case of the GP bus grant configuration, the portion of the internal (or private) bus address space that is mapped onto the expanded bus when the part is operating as either a GP or a UPC bus request device is determined by the expansion configuration selected. In the partial expansion bus requestor case, the resulting memory map is identical to that shown for the GP bus grant configuration in Figure 12. During the time the MK68HC200 is executing its programs from ROM and accessing internal RAM and I/O resources, the expanded bus is held in a tri-state condition. The bus arbitration logic within the MK68HC200 monitors each memory reference to detect external bus addresses (referenced in segments via the 16K byte DMA window). Whenever such an external reference occurs, the logic automatically holds the processor in a wait state as it proceeds to obtain mastership of the bus. When use of the system bus is obtained, the processor is allowed to continue the reference. This procedure is transparent to the programmer. In case of successive external references, the expanded bus is retained until an internal reference is encountered.

Finally, if the on-chip resources are insufficient to perform the control task in the bus requestor configuration, the internal bus address range (excluding on-chip RAM, I/O) may be mapped onto an external local bus, which is physically the same as the system bus but logically separated with bus buffers. This is the full expansion bus requestor configuration. The memory map for this configuration is shown in Figure 14. The bus arbitration sequence is performed only when the system bus is referenced through the DMA window. In this manner, the I/O subsystem is isolated from the host CPU.

When operating as a bus request device, it is possible to retain the external bus for an indefinite duration by using a bus lock feature. This will help facilitate the transfer of large blocks of data. Thus, the on-chip bus arbitration logic allows (with a minimum of hardware and software overhead) a maximum of concurrent processing in parallel, multiprocessing configurations. The bus lock feature may be used by the MK68HC200 in a bus grantor mode to keep any peripheral from gaining mastership of the bus.

In any of the GP expanded bus modes, the MK68HC200 may respond to peripheral devices on the expanded bus which generate an interrupt request on XI0. The MK68HC200 will obtain the XI0 interrupt vector number from the requesting peripheral on the bus during an interrupt acknowledge cycle. When responding to an interrupt on XI0, the MK68HC200 will wait for the bus arbitration logic to gain control of the bus and then asserts neither $\overline{HB}$ nor $\overline{LB}$ while asserting $\overline{AS}$ to signify that an interrupt acknowledge cycle is in progress. The X10 interrupt will be the lowest priority interrupt when operating in any of the GP expanded bus modes.

There is a user-programmable speed selection associated with the read and write cycles for both the UPC and GP mask option parts. A bit in the expanded bus control and status register, Port 15, allows the user to select either the standard or the fast read/write cycle on the expanded bus. The standard bus cycle is four clock periods, while the fast bus cycle is three clock periods.

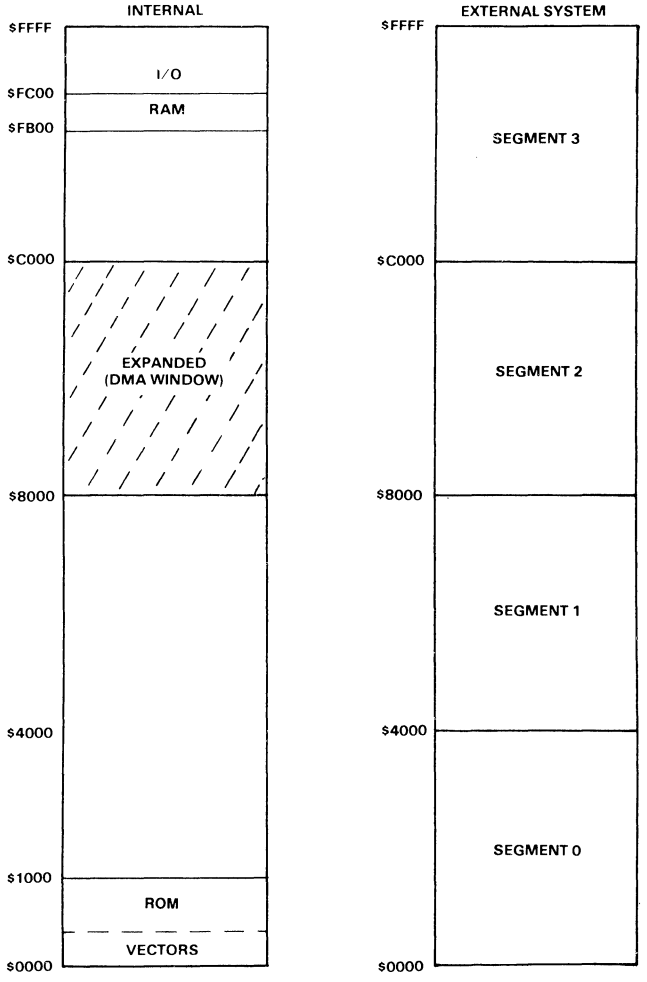


Figure 12. Partial Expansion Memory Map
(256 byte RAM, 4K byte ROM version shown)

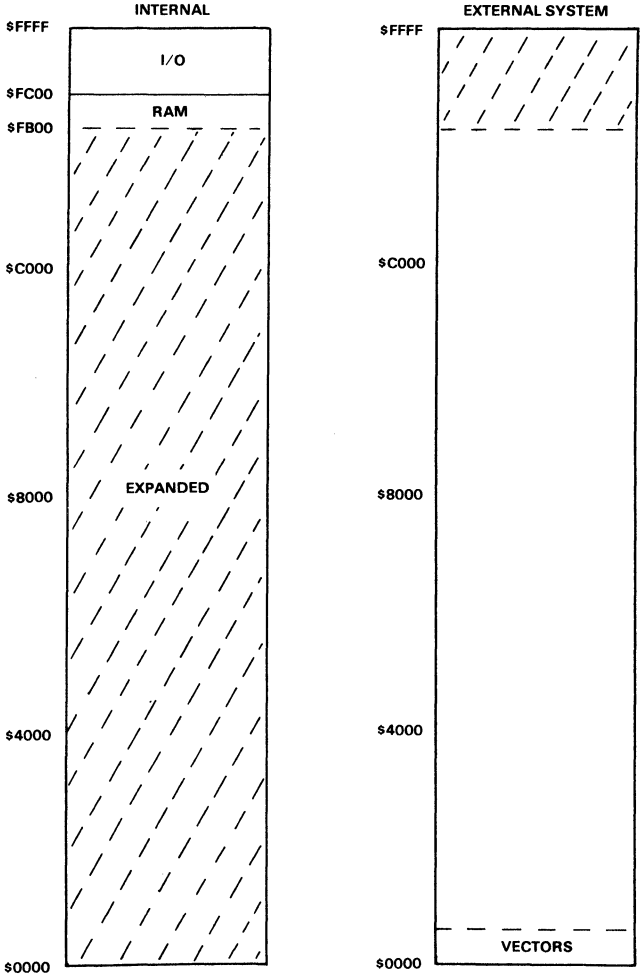


Figure 13. Full Expansion Bus Grantor Memory Map (256 byte RAM version shown)

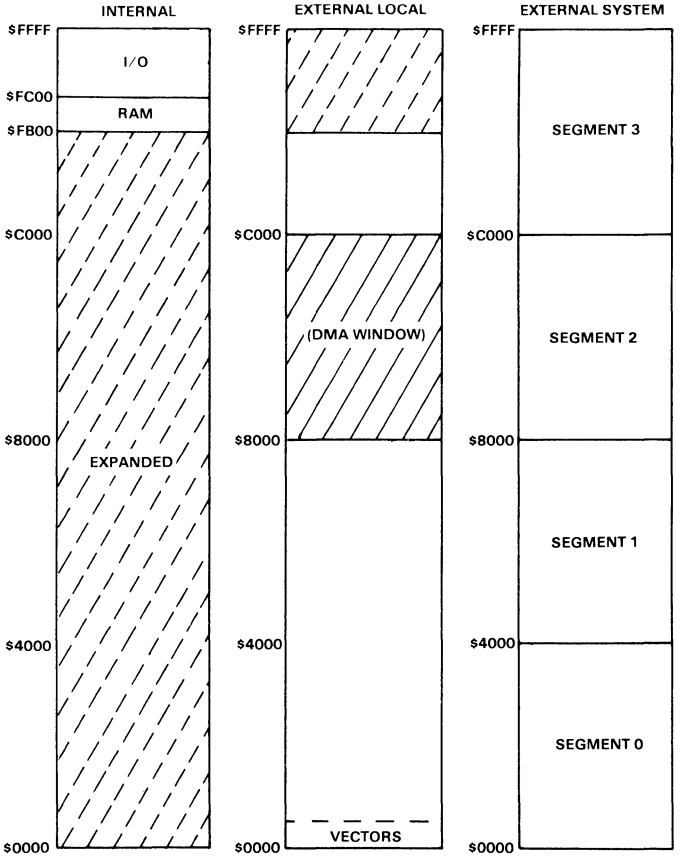


Figure 14. Full Expansion Bus Requestor Memory Map (256 byte RAM version shown)

EMULATOR VERSION

The emulator versions of the MK68HC200 are available in 84-pin, leadless or leaded chip carrier packages. Figure 15 illustrates the logical pinout of the emulator version. The emulator versions have no on-chip ROM, but instead include a second complete bus, referred to as the private bus. The private bus includes a multiplexed address/data bus as well as bus control signals. There are 22 pins associated with the private bus. All 40 I/O port pins that exist on the 52-pin versions are available to the user for configuration either as general purpose or special I/O pins, or as expanded bus pins.

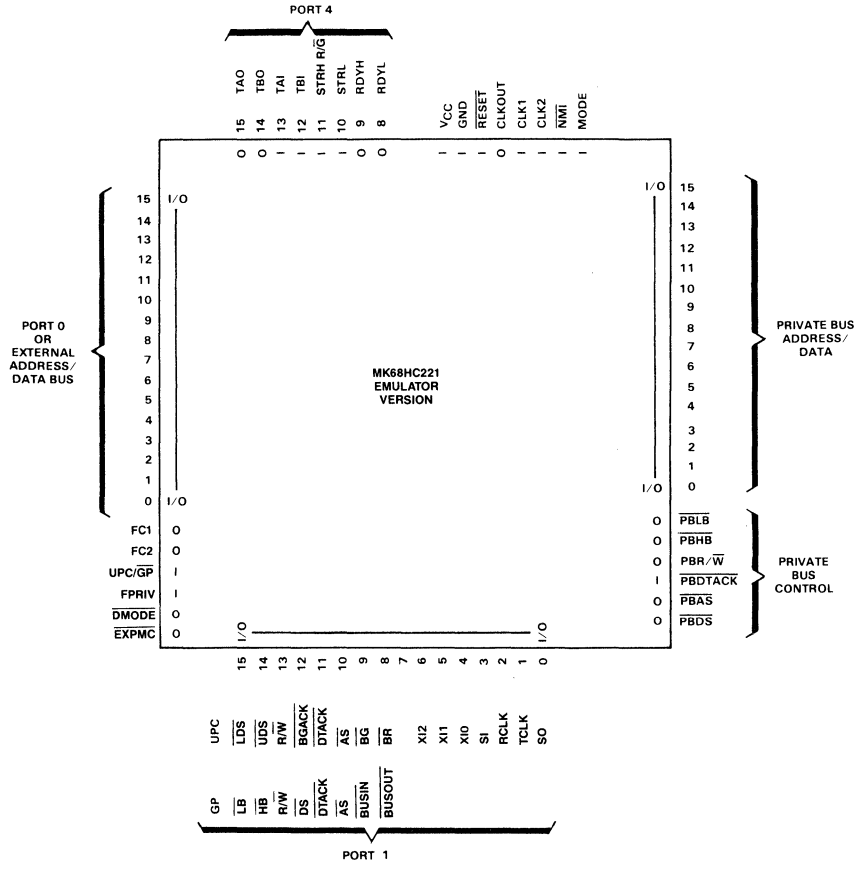


Figure 15. Logical Pinout for MK68HC221

PRIVATE BUS OPERATION

The address/data lines and control signals that constitute the private bus are functionally equivalent to the internal signals used to access internal resources on the ROM versions of the MK68HC200. Thus, the private bus may be used to interface to EPROM memory in emulating mask ROM versions of the MK68HC200. Alternately, any combination of ROM, RAM, and I/O may reside on the private bus.

The address that is generated on the private bus is identical to that which is internally generated for 48-pin versions. When the part is used in a configuration that supports system bus addressing through the DMA window, any references in this region of the memory map produce an address on the private bus identical to that specified by the programmer. In other words, the segment bits have no effect on the private bus address. The DMODE pin will go active during a reference to the DMA window. Write data appears on the private bus pins for all write operations, regardless of whether the reference is on-chip or off-chip. The MK68HC221 emulator version reads data from the private bus unless data is read from on-chip RAM, I/O, or the external bus formed by the Port 0 and Port 1 I/O pins.

The I/O port range of the memory map ($FC00-$FFFF) is actually subdivided into space which is exclusively reserved for on-chip I/O ($FC00-$FDFF) and space which is exclusively reserved for in-circuit-emulator use ($FE00-$FFFF). The user should ensure that no external devices reside in the in-circuit-emulator area.

The private bus interface is the same as that for the GP expanded bus. All read/write transfers made exclusively on the private bus are three clock periods, regardless of the state of the Fast/Standard (F/S) bus timing selection bit. In systems using the expanded bus, the user should be sure to tie the FPRIV pin low so that expanded bus operation is not effected. The user should ignore all activity on the private bus while accesses are in progress on the expanded bus. Care should also be taken that no external devices reside on the private bus in the memory space intended for expanded bus accesses.

There are six additional control pins available on the emulator version that are not on the ROM version. Five of these pins are meant to be used by the development system. FC1, FC2, DMODE, and EXPMC are used to define the memory cycle currently being executed. FPRIV will affect the memory cycle currently being executed. These signals are made available to simplify the design of the development system. Using these signals, the development system only has to interface to the private bus and not also to the expanded bus. The user might also be able to use these signals to simplify his design, however care should be taken when using FPRIV since this input will affect expanded bus memory cycles.

MNEMONIC	ADDITIONAL PIN FUNCTIONS FOR THE EMULATOR
UPC/GP	UPC/GP (input, active high and low). This pin is used to select either the UPC or GP control bus configuration for the expanded bus. (1 = UPC bus, 0 = GP bus). It is sampled only when reset is active
FC1, FC2	Function Code 1, Function Code 2 (outputs, active high and low). These pins (FC1 and FC2) define the memory cycle currently being executed. They are valid during the time private bus address strobe (PBAS) is active. The cycle types are interrupt, data, branch, and program fetch. The branch cycle is defined as the first program fetch after a branch occurs. A branch can occur as a result of a jump or call instruction, or an interrupt. For internal interrupts, the interrupt cycles are defined as the two writes to the stack and the read of the vector location which occur during the interrupt acknowledge routine. For external interrupts, the interrupt cycles are defined as the 3 cycles above plus the read of the vector number. The interrupt cycle is a special case of the data cycle. The function code pins are defined below.
DMODE	DMA Mode (output, active low). This pin goes low when the segment bits are being output on AD14 and AD15 on the expanded bus. (The address output on the private bus will not contain the segment bits.) DMODE is stable for the entire bus operation.
EXPMC	Expanded Memory Cycle (output, active low). This pin goes low when the expanded bus is being accessed. EXPMC is stable for the entire bus operation.
FPRIV	Force Private (input, active high). This pin is used to force the MK68HC200 to read data from the private bus when the address is actually located on the expanded bus. In normal operation this pin should be tied low and the expanded bus operation will be unaffected.

TYPE OF CYCLE	FC1	FC2
Interrupt	0	0
Data	0	1
Branch	1	0
Program Fetch	1	1

CRYSTAL SELECTION

The wide frequency range of crystals that can be chosen for the MK68HC200 offers the user a large degree of flexibility. To aid in the selection of a suitable crystal, the suggestions shown in Figure 16 should be considered by the user. The MK68HC200 offers an output pin that will provide a system clock signal at one-half of the crystal frequency.

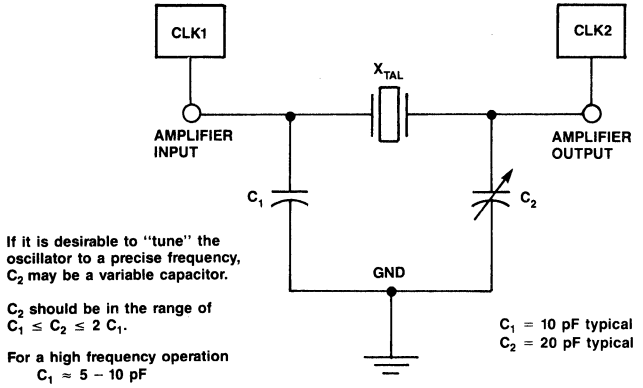


Figure 16. Crystal Connection

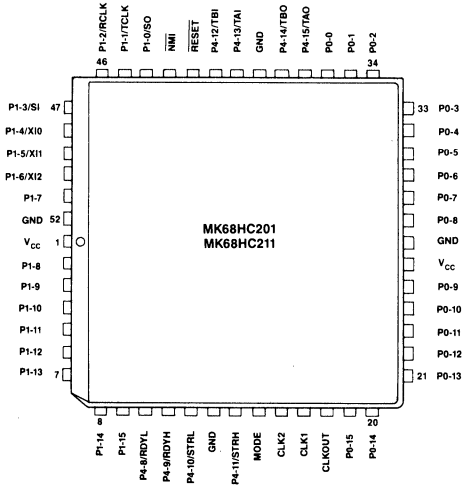
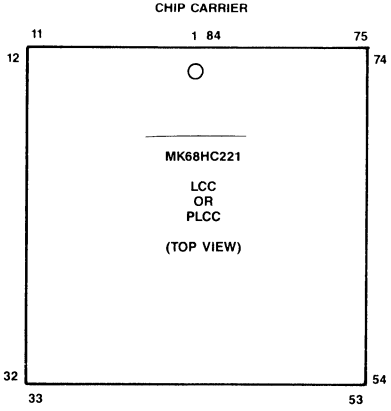


Figure 17a. MK68HC201, MK68HC211 Pin Assignment, ROM Version



LCC	FUNCTION	LCC	FUNCTION	LCC	FUNCTION	LCC	FUNCTION
1	P1-4/XIO	22	$\overline{\text{PBAS}}$	43	P0-7	64	PB-8
2	P1-5/XI1	23	$\overline{\text{PBDS}}$	44	P0-6	65	PB-9
3	P1-6/XI2	24	P4-8/RDYL	45	P0-5	66	PB-10
4	P1-7	25	P4-9/RDYH	46	P0-4	67	PB-11
5	P1-8	26	P4-10/STRL	47	P0-3	68	PB-12
6	P1-9	27	P4-11/STRH	48	P0-2	69	PB-13
7	P1-10	28	MODE	49	P0-1	70	PB-14
8	P1-11	29	CLK2	50	P0-0	71	PB-15
9	$\overline{\text{EXPMC}}$	30	CLK1	51	FC2	72	P4-13/TAI
10	VCC	31	CLKOUT	52	$\overline{\text{DMODE}}$	73	P4-12/TBI
11	NO CONNECT	32	FC1	53	FPRIV	74	VCC
12	GROUND	33	VCC	54	VCC	75	GROUND
13	UPC/$\overline{\text{GP}}$	34	GROUND	55	GROUND	76	P4-15/TAO
14	P1-12	35	P0-15	56	PB-0	77	P4-14/TBO
15	P1-13	36	P0-14	57	PB-1	78	$\overline{\text{RESET}}$
16	P1-14	37	P0-13	58	PB-2	79	$\overline{\text{NMI}}$
17	P1-15	38	P0-12	59	PB-3	80	P1-0/SO
18	$\overline{\text{PBLB}}$	39	P0-11	60	PB-4	81	P1-1/TCLK
19	$\overline{\text{PBHB}}$	40	P0-10	61	PB-5	82	P1-2/RCLK
20	PBR/$\overline{\text{W}}$	41	P0-9	62	PB-6	83	P1-3/SI
21	$\overline{\text{PBDTACK}}$	42	P0-8	63	PB-7	84	VCC

Figure 17b. MK68HC221 Pin Assignment, Emulator Version

ASSEMBLER DIRECTIVES

Directive	Function	Assembler Syntax		
DC	Define constant	[label:]	DC[.size]*1	expr {,expr}
DS	Define storage	[label:]	DS[.size]*1	expr
DUP	Duplicate constant block	[label:]	DUP[.size]*1	length, value
END	Program end		END	[start address]
EQU	Equate symbol value	label:	EQU	expr
FAIL	Programmer generated error		FAIL	expr
FORMAT	Format the source listing		FORMAT	
IDNT	Generate module ID	module_name:	IDNT	version, revision
LIST	Enable the assembly listing		LIST	
LLEN	Specify line length		LLEN	length
NOFORMAT	Do not format listing		NOFORMAT	
NOLIST	Disable assembly listing		NOLIST	
NOOBJ	Disable object code generation		NOOBJ	
NOPAGE	Suppress paging		NOPAGE	
OFFSET	Define Offsets		OFFSET	expr
OPT	Assembler output options		OPT	option*2 {, option}
ORG	Define absolute origin		ORG	expr
PAGE	Eject a page in the listing		PAGE	
REG	Define register list	reg_list_name:	REG[.size]	register list
SECTION	Define relocatable program section	[section_name:]	SECTION	number
SET	Set symbol value	label:	SET	expr
SPC	Space between source lines		SPC	number
TTL	Specify heading title string		TTL	title string
XDEF	External symbol definition		XDEF	symbol {, symbol}
XREF	External symbol reference		XREF	[sect no:] symbol - {,[sect no]: symbol}

NOTES:

1. .size = .B or .W (byte or word size)
2. Options for the OPT directive include:

CEX	Print DC expansions
NOCEX	Do not print DC expansions (default)
CL	Print conditional assembly directives (default)
NOCL	Do not print conditional assembly directives
CRE	Print cross-reference table
IMM.L	Forces immediate operands for arithmetic instructions ADD, SUB, DADD, and DSUB to use the long instruction form
IMM.S	Allows the assembler to select automatically the short form of the arithmetic instructions for small immediate values (0-15) (default)
MC	Print macro calls (default)
NOMC	Do not print macro calls
MD	Print macro definitions (default)
NOMD	Do not print macro definitions
MEX	Print macro expansions
NOMEX	Do not print macro expansions (default)
O	Create object module (default)
NOO	Do not create object module
STR	Print code generated by structured statements
NOSTR	Do not print code generated by structured statements (default)

GENERAL SYMBOL DEFINITIONS

SYMBOL	GENERAL SYMBOL DEFINITIONS
Rn	General Purpose Registers - D0-D7, A0-A5, SP, SR, DH0-DH7, DL0-DL7.
RPn	Register Pairs - D0-D1, D2-D3, D4-D5, D6-D7, A0-A1, A2-A3, A4-A5.
An	Address Registers - A0-A5, SP.
Pn	Ports - P0-P15, PH0-PH3, PL0-PL3.
CC	Condition Code - See Table.
d16	16-Bit Address Displacement Field In Words.
d13	13-Bit Address Displacement Field In Bytes.
d9	9-Bit Address Displacement Field In Bytes.
d8	8-Bit Address Displacement Field In Bytes.
#nx	Immediate Data Field - x Number of Bits.[1]
[s]	Size Bit - '1' = Word, '0' = Byte.
REGn	4-Bit Register Field - See Table.
[PORTn]	4-Bit Port Field - See Table.
[An]	3-Bit Address Register Field - See Table.
[PRTn]	3-Bit Port Field - See Table.
[RGn]	3-Bit Register Pair Field - See Table.
M	Register Mask Field - See Table.
[COND]	Condition Code Field - See Table.
[c3]	3-Bit Class Field - See Table.
[c2]	2-Bit Class Field - See Table
[c1]	1-Bit Class Field—See Table
a	Address Field - 16 Bits.
#	Immediate Data Field.
n	3-Bit Shift Field - $2 \leq n \leq 7$.
b#	4-Bit Bit Select Field.
d	Displacement Field.
.B	Byte Attribute.
.W	Word Attribute.
.L	Long Attribute.
.S	Short Attribute.
[ ]	Optional Field.

NOTE

1. When using the byte format of an instruction with a 16-bit immediate data field, both the high and low byte of the data field must contain the same 8-bit data.

FIELD DEFINITIONS

REGn 4-Bit Register Map					
Register		Bit Field	Register		Bit Field
D0	DH0	0 0 0 0	A0	DL0	1 0 0 0
D1	DH1	0 0 0 1	A1	DL1	1 0 0 1
D2	DH2	0 0 1 0	A2	DL2	1 0 1 0
D3	DH3	0 0 1 1	A3	DL3	1 0 1 1
D4	DH4	0 1 0 0	A4	DL4	1 1 0 0
D5	DH5	0 1 0 1	A5	DL5	1 1 0 1
D6	DH6	0 1 1 0	SP	DL6	1 1 1 0
D7	DH7	0 1 1 1	SR	DL7	1 1 1 1

PORTn 4-Bit Port Map				
Port		Bit Field	Port	Bit Field
P0	PH0	0 0 0 0	P8	1 0 0 0
P1	PL0	0 0 0 1	P9	1 0 0 1
P2	PH1	0 0 1 0	P10	1 0 1 0
P3	PL1	0 0 1 1	P11	1 0 1 1
P4	PH2	0 1 0 0	P12	1 1 0 0
P5	PL2	0 1 0 1	P13	1 1 0 1
P6	PH3	0 1 1 0	P14	1 1 1 0
P7	PL3	0 1 1 1	P15	1 1 1 1

An 3-Bit Addr Reg Map	
Register	Bit Field
A0	0 0 0
A1	0 0 1
A2	0 1 0
A3	0 1 1
A4	1 0 0
A5	1 0 1
SP	1 1 0

PTRn 3-Bit Port Map	
Port	Bit Field
PH0	0 0 0
PL0	0 0 1
PH1	0 1 0
PL1	0 1 1
PH2	1 0 0
PL2	1 0 1
PH3	1 1 0
PL3	1 1 1

RGn 3-Bit Reg Pair Map	
Register	Bit Field
D0-D1	0 0 0
D2-D3	0 0 1
D4-D5	0 1 0
D6-D7	0 1 1
A0-A1	1 0 0
A2-A3	1 0 1
A4-A5	1 1 0

M—REGISTER MASK MAP FOR MOVEM, PUSHM, AND POPM																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Inc Word	SR	SP	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
Dec Word	D0	D1	D2	D3	D4	D5	D6	D7	A0	A1	A2	A3	A4	A5	SP	SR
Inc Byte	DH7	DL7	DH6	DL6	DH5	DL5	DH4	DL4	DH3	DL3	DH2	DL2	DH1	DL1	DH0	DL0
Dec Byte	DL0	DH0	DL1	DH1	DL2	DH2	DL3	DH3	DL4	DH4	DL5	DH5	DL6	DH6	DL7	DH7

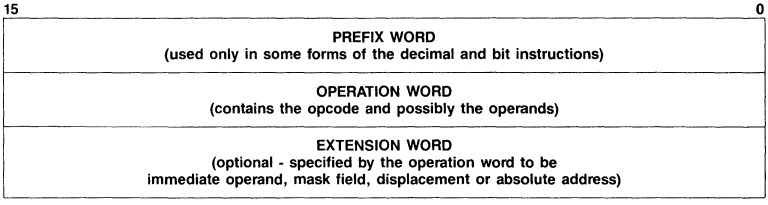
COND CONDITION CODE TABLE			
Condition Code	**Bit Field**	**Description**	**Test**
Z EQ	0 0 0 0	Zero Equal	Z
MI	0 0 0 1	Minus	N
LO[2] CS	0 0 1 0	Lower Carry Set	C
VS	0 0 1 1	Overflow Set	V
GE[2]	0 1 0 0	Greater than or Equal	$\overline{N}$ .EOR. $\overline{V}$
GT[2]	0 1 0 1	Greater than	$\overline{Z}$ .AND. ($\overline{N}$ .EOR. $\overline{V}$)
HI[2]	0 1 1 0	Higher	$\overline{C}$ .AND. $\overline{Z}$
F[1]	0 1 1 1	False	Always False
NE NZ	1 0 0 0	Not Equal Not Zero	$\overline{Z}$
PL	1 0 0 1	Plus	$\overline{N}$
HS[2] CC	1 0 1 0	Higher or Same Carry Clear	$\overline{C}$
VC	1 0 1 1	Overflow Clear	$\overline{V}$
LT[2]	1 1 0 0	Less than	N .EOR. V
LE[2]	1 1 0 1	Less than or Equal	Z .OR. (N .EOR. V)
LS[2]	1 1 1 0	Lower or Same	C .OR. Z
T[1]	1 1 1 1	True	Always True

NOTES:
1. The assembler does not recognize the T and F condition codes.
2. LT, LE, GT, and GE are used for unsigned conditions; LO, LS, HI, and HS are for unsigned conditions.

INSTRUCTION CLASS FIELDS									
c3 - 3-Bit Field			c2 - 2-Bit Field			c1 - 1-Bit Field			
Bit Field	Shift Instr	Bit[1] Instr	Bit Field	Arith[2] Instr	Logical Instr	Bit Field	Arith[2] Instr	Test Instr	Neg[2] Instr
0 0 0	ROR	BSET	0 0	ADDC	OR	0	ADD	TESTN	NEGC
0 0 1	ROL	BCHG	0 1	SUBC	EOR	1	SUB	TEST	NEG
0 1 0	RORC	BCLR	1 0	ADD	AND				
0 1 1	ROLC	BTST	1 1	SUB	—				
1 0 0	ASR	—							
1 0 1	ASL	—							
1 1 0	LSR	—							
1 1 1	—	BEXG							

NOTES:
1. The bit fields do not apply to bit instructions using a port operand.
2. These fields also apply to BCD instructions.

INSTRUCTION FORMAT



INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
ADD ADDC SUB SUBC	B [.W]	Ry,Rx	2 REGx s c2 0 REGy	1	—	3	ADD: Src + Dst → Dst	* * * *
	B [.W]	(Ay),Rx	2 REGx s c2 1 0 Ay	1	—	6	ADDC: Src + Dst + C → Dst	
	B [.W]	d16(Ay),Rx	2 REGx s c2 1 1 Ay	2	d	9	SUB: Dst − Src → Dst	
	B [.W]	Addr,Rx	2 REGx s c2 1 F	2	a	9	SUBC: Dst − Src − C → Dst	
	B [.W]	#n16,Rx	2 REGx s c2 1 7	2	#	6	Note: For addressing modes #n,Rx and #n,(Ax) with the ADD and SUB instructions, the assembler uses the short version for immediate values ≤ 4 bits.	
	B [.W]	Ry,(Ax)	3 0 Ax s c2 0 REGy	1	—	9		
	B [.W]	(Ax),(Ay)	3 0 Ax s c2 1 0 Ay	1	—	12		
	B [.W]	#n16,(Ax)	3 0 Ax s c2 1 7	2	#	12		
	B [.W]	(Ax) + ,(Ay) +	3 1 Ax s c2 1 0 Ay	1	—	12		
	B [.W]	#n16,(Ax) +	3 1 Ax s c2 1 7	2	#	12		
	B [.W]	− (Ax), − (Ay)	3 1 Ax s c2 1 1 Ay	1	—	12		
	B [.W]	#n16, − (Ax)	3 1 Ax s c2 1 F	2	#	12		
	B [.W]	Ry,d16(Ax)	3 1 Ax s c2 0 REGy	2	d	12		
	B [.W]	Ry,Addr	3 F s c2 0 REGy	2	a	12		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
ADD SUB	B [.W]	#n4,Rx	B REGx s 1 c1 0 #	1	—	3	ADD: Src + Dst → Dst	* * * *
	B [.W]	#n4,(Ax)	B 0 Ax s 1 c1 1 #	1	—	9		
	B [.W]	#n4,d16(Ax)	B 1 Ax s 1 c1 1 #	2	d	12	SUB: Dst − Src → Dst	
	B [.W]	#n4,Addr	B F s 1 c1 1 #	2	a	12		
AND EOR OR	B [.W]	Ry,Rx	6 REGx s c2 0 REGy	1	—	3	AND: Src .AND. Dst → Dst	* * 0 0
	B [.W]	Py,Rx	5 REGx s c2 0 PORTy	1	—	6		
	B [.W]	(Ay),Rx	6 REGx s c2 1 0 Ay	1	—	6	EOR: Src .EOR. Dst → Dst	
	B [.W]	d16(Ay),Rx	6 REGx s c2 1 1 Ay	2	d	9		
	B [.W]	Addr,Rx	6 REGx s c2 1 F	2	a	9	OR: Src .OR. Dst → Dst	
	B [.W]	#n16,Rx	6 REGx s c2 1 7	2	#	6		
	B [.W]	Ry,Px	4 PORTx s c2 0 REGy	1	—	9		
	[.W]	#n16,Px	5 PORTx 1 c2 1 9	2	#	12		
	B [.W]	Ry,(Ax)	7 0 Ax s c2 0 REGy	1	—	9		
Continued on next page	B [.W]	(Ax),(Ay)	7 0 Ax s c2 1 0 Ay	1	—	12		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
AND EOR OR (cont.)	.B [.W]	#n16,(Ax)	7 0 Ax s c2 1 7	2	#	12	AND: Src .AND. Dst → Dst	* * 0 0
	.B [.W]	(Ax) + ,(Ay) +	7 1 Ax s c2 1 0 Ay	1	—	12		
	.B [.W]	#n16,(Ax) +	7 1 Ax s c2 1 7	2	#	12	EOR: Src .EOR. Dst → Dst	
	.B [.W]	-(Ax), -(Ay)	7 1 Ax s c2 1 1 Ay	1	—	12		
	.B [.W]	#n16, -(Ax)	7 1 Ax s c2 1 F	2	#	12	OR: Src .OR. Dst → Dst	
	.B [.W]	Ry,d16(Ax)	7 1 Ax s c2 0 REGy	2	d	12		
	.B [.W]	Ry,Addr	7 F s c2 0 REGy	2	a	12		
ASL ASR LSR ROL ROLC ROR RORC	.B [.W]	[#1],Rx	B REGx s 0 c3 0 0 1	1	—	3	Dst → Dst SHIFT: C ◄ ASL ◄ 0 0 ► LSR ► C ► ASR ► C ROTATE: C ◄ ROL ◄ C ◄ ROLC ◄ ► ROR ► C ► RORC ► C *NOTE: 2 ≤ n ≤ 7	* * * * NOTE: The ASR, LSR, ROR, and RORC instructions will clear the V status bit.
	.B [.W]	#n3,Rx	B REGx s 0 c3 n	1	—	4 + n*		
	.B [.W]	(Ax)	B 0 Ax s 0 c3 0 0 0	1	—	9		
	.B [.W]	d16(Ax)	B 1 Ax s 0 c3 0 0 0	2	d	12		
	.B [.W]	Addr	B F s 0 c3 0 0 0	2	a	12		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
BCHG BCLR BEXG BSET BTST	—	#n4,Rx	4 REGx c3 1 b#	1	—	3	BCHG: Dst(b#) → C 1 — Dst(b#) → Dst(b#)	* * 0 * If Dst(b#) is in the low byte of the SR,then the other SR bits are unchanged.
	—	#n4,(Ax)	5 0 Ax c3 1 b#	1	—	9*		
	—	#n4,d16(Ax)	5 1 Ax c3 1 b#	2	d	12*	BCLR: Dst(b#) → C 0 → Dst(b#)	
	—	#n4,Addr	5 F c3 1 b#	2	a	12*		
	—	Ry,Rx	0 REGy F 7	2	—	6	BEXG: Dst(b#) ↔ C	
			4 REGx c3 1 0					
	—	Ry,(Ax)	0 REGy F 7	2	—	12*	BSET: Dst(b#) → C 1 → Dst(b#)	
			5 0 Ax c3 1 0					
	—	Ry,d16(Ax)	0 REGy F 7	3	d	15*	BTST: Dst(b#) → C	
			5 1 Ax c3 1 0					
	—	Ry,Addr	0 REGy F 7	3	a	15*	*NOTE: The BTST instruction executes in 3 less clock cycles.	
			5 F c3 1 0					

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
BCHG	—	#n4,Px	4 PORTx B b#	1	—	9	Dst(b#) → C 1 — Dst(b#) → Dst(b#)	* * 0 *
	—	Ry,Px	0 REGy F 7	2	—	12		
			4 PORTx B 0					
BCLR	—	#n4,Px	4 PORTx D b#	1	—	9	Dst(b#) → C 0 → Dst(b#)	
	—	Ry,Px	0 REGy F 7	2	—	12		
			4 PORTx D 0					
BEXG	—	#n4,Px	7 PORTx E b#	1	—	9	Dst(b#) → C	
	—	Ry,Px	0 REGy F 7	2	—	12		
			7 PORTx E 0					
BSET	—	#n4,Px	4 PORTx 9 b#	1	—	9	Dst(b#) → C 1 → Dst(b#)	
	—	Ry,Px	0 REGy F 7	2	—	12		
			4 PORTx 9 0					
BTST Continued on next page	—	#n4,Px	7 PORTx 6 b#	1	—	6	Dst(b#) → C	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
BTST (cont.)	—	Ry,Px	0 REGy F 7	2	—	9	Dst (b#) → C	* * 0 *
			7 PORTx 6 0					
CALLA	[.L]	(Ax) (unconditional)	5 0 Ax D F	1	—	9	PC + 2 → –(SP) (Ax) → PC	— — — —
	[.L]	Addr (unconditional)	5 F B F	2	a	9	PC + 4 → –(SP) Addr → PC	
	[.L]	CC, Addr	5 COND B F	2	a	F:6 T:12	If COND is true, PC + 4 → –(SP); Addr → PC	
							NOTE: The initial PC value is the location of the CALLA instruction.	
CALLR	[.S]	d13 (unconditional)	F d	1	—	10	PC + 2 → –(SP) PC + 4 + 2•(d) → PC	— — — —
	.L	d16 (unconditional)	5 F 9 F	2	d	9	PC + 4 → –(SP) PC + 4 + d → PC	
	.L	CC, d16	5 COND 9 F	2	d	F:6 T:12	If COND is true, PC + 4 → –(SP); PC + 4 + d → PC	
							NOTE: The initial PC value is the location of the CALLR instruction. The displacement, d13, is in signed magnitude representation, and the displacement, d16, is in two's complement representation.	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
CLR	.B [.W]	Rx	1 REGx s 0 1 1 7	1	—	3	0 → Dst	— — — —
	.B [.W]	Px	1 PORTx s 0 1 0 7	1	—	6		
	.B [.W]	(Ax)	1 0 Ax s 0 1 1 F	1	—	6		
	.B [.W]	d16(Ax)	1 1 Ax s 0 1 1 F	2	d	9		
	.B [.W]	Addr	1 F s 0 1 1 F	2	a	9		
CMP	.B [.W]	Ry,Rx	6 REGx s 1 1 0 REGy	1	—	3	Dst — Src	* * * *
	.B [.W]	Py,Rx	5 REGx s 1 1 0 PORTy	1	—	6		
	.B [.W]	(Ay),Rx	6 REGx s 1 1 1 0 Ay	1	—	6		
	.B [.W]	(Ay) + ,Rx	4 REGx s 1 1 0 0 Ay	1	—	6		
	.B [.W]	– (Ay),Rx	4 REGx s 1 1 0 1 Ay	1	—	6		
	.B [.W]	d16(Ay),Rx	6 REGx s 1 1 1 1 Ay	2	d	9		
	.B [.W]	Addr,Rx	6 REGx s 1 1 1 F	2	a	9		
	.B	#n8,Rx	C REGx #	1	—	3		
Continued on next page	.[W]	#n16,Rx	6 REGx F 7	2	#	6		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
CMP (cont.)	.B	#n8,Px	D 0 PRTx #	1	—	6	Dst — Src	* * * *
	[.W]	#n16,Px	4 PORTx E F	2	#	9		
	.B [.W]	(Ax),(Ay)	7 0 Ax s 1 1 1 0 Ay	1	—	9		
	.B [.W]	#n16,(Ax)	7 0 Ax s 1 1 1 7	2	#	9		
	.B [.W]	(Ax) + ,(Ay) +	7 1 Ax s 1 1 1 0 Ay	1	—	9		
	.B [.W]	#n16,(Ax) +	7 1 Ax s 1 1 1 7	2	#	9		
	.B [.W]	– (Ax), – (Ay)	7 1 Ax s 1 1 1 1 Ay	1	—	9		
	.B [.W]	#n16, – (Ax)	7 1 Ax s 1 1 1 F	2	#	9		
DADD DADDC DSUB DSUBC	.B [.W]	Ry,Rx	A REGx s c2 0 REGy	1	—	3	DADD: BCD [Src + Dst → Dst]	U * U *
	.B [.W]	(Ay),Rx	A REGx s c2 1 0 Ay	1	—	6	DADDC: BCD [Src + Dst + C → Dst]	
	.B [.W]	d16(Ay),Rx	A REGx s c2 1 1 Ay	2	d	9	DSUB: BCD [Dst – Src → Dst]	
	.B [.W]	Addr,Rx	A REGx s c2 1 F	2	a	9		
Continued on next page	.B [.W]	#n16,Rx	A REGx s c2 1 7	2	#	6	DSUBC: BCD [Dst – Src – C → Dst]	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
DADD DADDC DSUB DSUBC (cont)	B [W]	Ry,(Ax)	0 F F F	2	—	12		U * U *
			3 0 Ax s c2 0 REGy					
	B [W]	(Ax),(Ay)	0 F F F	2	—	15	NOTE: For addressing modes #n,Rx and #n,(Ax) with the DADD and DSUB instructions, the assembler uses the short version for immediate values ≤ 4 bits	
			3 0 Ax s c2 1 0 Ay					
	B [W]	#n16,(Ax)	0 F F F	3	#	15		
			3 0 Ax s c2 1 /					
	B [W]	(Ax) + ,(Ay) +	0 F F F	2	—	15		
			3 1 Ax s c2 1 0 Ay					
	B [W]	#n16,(Ax) +	0 F F F	3	#	15		
			3 1 Ax s c2 1 7					
	B [W]	- (Ax), - (Ay)	0 F F F	2	—	15		
			3 1 Ax s c2 1 1 Ay					
	B [W]	#n16, - (Ax)	0 F F F	3	#	15		
Continued on next page			3 1 Ax s c2 1 F					

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
DADD DADDC DSUB DSUBC (cont)	B [W]	Ry,d16(Ax)	0 F F F	3	d	15		U * U *
			3 1 Ax s c2 0 REGy					
	B [W]	Ry,Addr	0 F F F	3	a	15		
			3 F s c2 0 REGy					
DADD DSUB	.B [W]	#n4,Rx	0 F F F	2	—	6	DADD: BCD: [Src + Dst → Dst]	U * U *
			B REGx s 1 c1 0 #					
	.B [W]	#n4,(Ax)	0 F F F	2	—	12	DSUB: BCD: [Dst - Src → Dst]	
			B 0 Ax s 1 c1 1 #					
	B [W]	#n4,d16(Ax)	0 F F F	3	d	15		
			B 1 Ax s 1 c1 1 #					
	.B [W]	#n4,Addr	0 F F F	3	a	15		
			B F s 1 c1 1 #					
DI	—	—	4 F 5 7	1	—	3	0 → L1E : Disable Interrupts	— — — —

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
DIVU	—	Rx,RPy	5 | REGx | D | RGy | 0	1	—	F:6 T:23	RPy(even),RPy(odd)/Rx → RPy(even) REM → RPy(odd) MSW → even; LSW → odd	— — * —
DJNZ	.B [.W]	Rx,d8	9 | REGx | s | d	1	—	F:6 T:9	Rx − 1 → Rx; If Rx ≠ 0 then PC+4−[2•(d)] → PC NOTE: The initial PC value is the location of the DJNZ instruction. The displacement value, d, is a magnitude.	— — — —
DNEG DNEGC	.B [.W]	Ry	0 | F | F | F	2	—	6	DNEG: BCD: [0 − Dst → Dst]	U * U *
			3 | 7 | s c1 1 0 | REGy					
	.B [.W]	(Ay)	0 | F | F | F	2	—	12	DNEGC: BCD: [0−Dst−C → Dst]	
			3 | 7 | s c1 1 1 0 | Ay					
	.B [.W]	d16(Ay)	0 | F | F | F	3	d	15		
			3 | 7 | s c1 1 1 1 | Ay					
	.B [.W]	Addr	0 | F | F | F	3	a	15		
			3 | 7 | s c1 1 1 | F					
DSUB DSUBC		see DADD (page 15)						

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
EI	—	—	4 | F | 1 | 7	1	—	3	1 → L1E Enable Interrupts	— — — —
EOR		see AND (page 9)						
EXG	.B [.W]	Rx,Ry	0 | REGx | s 1 0 0 | REGy	1	—	4	Src ↔ Dst	— — — —
	.B [.W]	Rx,(Ay)	0 | REGx | s 1 0 1 0 | Ay	1	—	9		
	.B [.W]	Rx,(Ay)+	0 | REGx | s 1 1 1 0 | Ay	1	—	9		
	.B [.W]	Rx, −(Ay)	0 | REGx | s 1 1 1 1 | Ay	1	—	9		
	.B [.W]	Rx,d16(Ay)	0 | REGx | s 1 0 1 1 | Ay	2	d	12		
	.B [.W]	Rx,Addr	0 | REGx | s 1 0 1 | F	2	a	12		
EXT	—	Rx	0 | REGx | D | 7	1	—	3	SIGN EXTEND: SignBit(RLx) → RHx	— — — —
HALT	—	—	1 | 7 | 3 | F	1	—	—	PC + 2 → PC; Fetch continuously	— — — —
IDLE	—	—	1 | 7 | 1 | F	1	—	—	PC + 2 → PC; Pause; CPU clocks off; I/O still active	— — — —
JMPA	[.L]	(Ax) (unconditional)	5 | 1 Ax | D | F	1	—	6	(Ax) → PC	— — — —
	[.L]	Addr (unconditional)	5 | F | B | 7	2	a	6	Addr → PC	
	[.L]	CC,Addr	5 | COND | B | 7	2	a	F:6 T:9	If COND is true, Addr → PC	
LINK	—	Ax,#d16	5 | 1 Ax | 9 | D	2	d	9	Ax → −(SP); SP → Ax; SP+d → SP	— — — —

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
JMPR	[.S]	d9 (unconditional)	8 F d	1	—	7	PC + 4 + 2•(d) → PC	— — — —
	[.S]	CC,d9	8 COND d	1	—	F:4 T:7	If COND is true, PC + 4 + 2•(d) → PC	
	L	d16 (unconditional)	5 F 9 7	2	d	9	PC + 4 + d → PC	
	L	CC,d16	5 COND 9 7	2	d	F:6 T:9	If COND is true, PC + 4 + d → PC	
							NOTE: The initial PC value is the location of the JMPR instruction. The displacement, d16, is in two's complement representation, and the displacement, d9, is in signed magnitude representation.	
LIBA LIWA	—	#d16(Rx),Az	1 REGx s 1 Az 1 1 1	2	#	6	Rx[•2] + d → Az	— — — —
	—	#d16(Ay,Rx),Az	1 REGx s 1 Az Ay	2	#	6	Ay + (Rx[•2]) + d → Az	
							NOTE: [•2] used for LIWA only. REGx and Az must not refer to same register when using 2nd addressing mode	
LSR		see ASL (page 10)						
MOVE	.B [.W]	Ry,Rx	0 REGx s 0 0 0 REGy	1	—	3	Src → Dst	— — — —
	.B [.W]	Py,Rx	0 REGx s 0 1 1 PORTy	1	—	6		
Continued on next page	.B [.W]	(Ay),Rx	0 REGx s 0 0 1 0 Ay	1	—	6		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source = Src Destination = Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
MOVE (cont.)	.B [.W]	(Ay) + ,Rx	1 REGx s 0 1 1 0 Ay	1	—	6	Src → Dst	— — — —
	.B [.W]	- (Ay),Rx	1 REGx s 0 1 1 1 Ay	1	—	6		
	.B [.W]	d16(Ay),Rx	0 REGx s 0 0 1 1 Ay	2	d	9		
	.B [.W]	Addr,Rx	0 REGx s 0 0 1 F	2	a	9		
	.B	#n8,Rx	E REGx #	1	—	3		
	[.W]	#n16,Rx	0 REGx 9 7	2	#	6		
	.B [.W]	Ry,Px	0 PORTx s 0 1 0 REGy	1	—	6		
	.B [.W]	(Ay) + ,Px	0 PORTx s 1 1 0 0 Ay	1	—	9		
	.B	#n8,Px	D 1 PRTx #	1	—	6		
	[.W]	#n16,Px	0 PORTx E 7	2	#	9		
	.B [.W]	Ry,(Ax)	1 0 Ax s 0 0 0 REGy	1	—	6		
	.B [.W]	(Ax),(Ay)	1 0 Ax s 0 0 1 0 Ay	1	—	9		
	.B [.W]	#n16,(Ax)	1 0 Ax s 0 0 1 7	2	#	9		
Continued on next page	.B [.W]	Rx,(Ay) +	1 REGx s 0 1 0 0 Ay	1	—	6		

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
MOVE (cont.)	.B [.W]	(Ax)+,(Ay)+	1 1 Ax s 0 0 1 0 Ay	1	—	9	Src → Dst	— — — —
	.B [.W]	#n16,(Ax)+	1 1 Ax s 0 0 1 7	2	#	9		
	.B [.W]	Rx,–(Ay)	1 REGx s 0 1 0 1 Ay	1	—	6		
	.B [.W]	Px,–(Ay)	0 PORTx s 1 1 0 1 Ay	1	—	9		
	.B [.W]	–(Ax),–(Ay)	1 1 Ax s 0 0 1 1 Ay	1	—	9		
	.B [.W]	#n16,–(Ax)	1 1 Ax s 0 0 1 F	2	#	9		
	.B [.W]	Ry,d16(Ax)	1 1 Ax s 0 0 0 REGy	2	d	9		
	.B [.W]	Ry,Addr	1 F s 0 0 0 REGy	2	a	9		
MOVEM	.B [.W]	(Ay)+,REGLIST	1 7 s 0 0 1 0 Ay	2	M	9+3n	(Ay)+ → REGLIST	— — — —
	.B [.W]	–(Ay),REGLIST	1 7 s 0 0 1 1 Ay	2	M	9+3n	–(Ay) → REGLIST	
	.B [.W]	REGLIST,(Ay)+	1 F s 0 0 1 0 Ay	2	M	7+3n	REGLIST → (Ay)+	
	.B [.W]	REGLIST,–(Ay)	1 F s 0 0 1 1 Ay	2	M	7+3n	REGLIST → –(Ay)	
							NOTE: A minimum of 2 registers must be specified and may be specified in any order.	

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
MULS	—	Rx,RPy	5 REGx B RGy 0	1	—	21	Rx * RPy(even) → RPy(even), RPy(odd) MSW → even; LSW → odd	— — — —
MULU	—	Rx,RPy	5 REGx 9 RGy 0	1	—	21	Rx * RPy(even) → RPy(even), RPy(odd) MSW → even; LSW → odd	— — — —
NEG NEGC	.B [.W]	Ry	3 7 s ct 1 0 REGy	1	—	3	2's Complement	* * * *
	.B [.W]	(Ay)	3 7 s ct 1 1 0 Ay	1	—	9	NEG: 0 – Dst → Dst	
	.B [.W]	d16(Ay)	3 7 s ct 1 1 1 Ay	2	d	12	NEGC: 0 – Dst – C → Dst	
	.B [.W]	Addr	3 7 s ct 1 1 F	2	a	12		
NOP	—	—	0 0 0 0	1	—	3	PC + 2 → PC	— — — —
NOT	.B [.W]	Ry	3 7 s 0 0 0 REGy	1	—	3	1's Complement	* * 0 0
	.B [.W]	(Ay)	3 7 s 0 0 1 0 Ay	1	—	9	NOT(Dst) → Dst	
	.B [.W]	d16(Ay)	3 7 s 0 0 1 1 Ay	2	d	12		
	.B [.W]	Addr	3 7 s 0 0 1 F	2	a	12		
OR		see AND (page 9)						

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
POP	—	Rx	1 REGx B 6	1	—	6	(SP)+ → Rx	— — — —
	—	Px	0 PORTx E 6	1	—	9	(SP)+ → Px	
	—	(Ay)+	1 E 9 0 Ay	1	—	9	(SP)+ → (Ay)+	
POPM	—	REGLIST	1 7 9 6	2	M	9+3n	(SP)+ → REGLIST NOTE: A minimum of 2 registers must be specified and may be specified in any order	— — — —
PUSH	—	Rx	1 REGx A E	1	—	6	Rx → -(SP)	— — — —
	—	Px	0 PORTx E E	1	—	9	Px → -(SP)	
	—	-(Ay)	1 1 Ax 9 E	1	—	9	-(Ax) → -(SP)	
	—	#n16	1 E 9 F	2	#	9	#<data> → -(SP)	
PUSHM	—	REGLIST	1 F 9 E	2	M	7+3n	REGLIST → -(SP) NOTE: A minimum of 2 registers must be specified and may be specified in any order.	— — — —
RET	—	—	5 6 D 7	1	—	9	(SP)+ → PC	— — — —
	—	Ax	5 0 Ax D 7	1	—	9	((Ax))+ → PC	
RETI	—	—	5 F D 7	1	—	12	(SP)+ → SR (SP)+ → PC	* * * *

INSTRUCTION MNEMONIC	ATTR SIZE	OPERAND ASSEMBLER SYNTAX Source=Src Destination=Dst Src , Dst	OPERATION WORD 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	WORDS	EXTENSION	CYCLES	OPERATION	STATUS FLAGS N Z V C
ROL ROLC ROR RORC		see ASL (page 10)						
STOP	—	—	1 7 9 F	1	—	—	Pause; CPU & I/O CLOCKS OFF	— — — —
SUB SUBC		see ADD (page 8)						
TEST TESTN	.B [.W]	#n16,Ry	7 7 s c1 0 0 REGy	2	#	6	TEST: Src .AND. Dst	* * 0 0
	.B [.W]	#n16,Py	7 F s c1 0 1 PORTy	2	#	9	TESTN: Src .AND. .NOT.(Dst)	
	.B [.W]	#n16,(Ay)	7 7 s c1 0 1 0 Ay	2	#	9		
UNLK	—	Ax	5 1 Ax 9 5	1	—	6	Ax → SP; (SP) + → Ax	— — — —

The following symbols are used to describe the state of the Status Register flags:
* Set according to result of operation.
0 Cleared.
1 Set.
— Not affected.
U Undefined.

Port Map

Port		Addr	Description	15	14	13	12	11	10	9	8
P0	PH0 PL0	$FC00	16 External I/O or Addr/Data Bus In Expanded Bus Mode								
P1	PH1 PL1	$FC02	16 Ext I/O, Ext Interrupts, Serial Port I/O, Bus Control	LB LDS	HB UDS	R/W	DS BGACK	DTACK	AS	BUSIN BG	BUSOUT BR
P2	PH2 PL2	$FC04	Reserved	← RESERVED →							
P3	PH3 PL3	$FC06	Serial Receive and Serial Transmit Buffers	RECEIVE BUFFER							
P4		$FC08	8 External I/O or Timer and Port 0 Handshake	TAO	TBO	TAI	TBI	STRH R/G	STRL	RDYH	RDYL
P5		$FC0A	Reserved	← RESERVED →							
P6		$FC0C	Reserved	← RESERVED →							
P7		$FC0E	Interrupt Latch Register	RES	NMI	SPARE	XI2I	STRLI	TAOI	TAII	STRHI
P8		$FC10	Interrupt Mask Register	RESERVED		SPARE	XI2M	STRLM	TAOM	TAIM	STRHM
P9		$FC12	Serial I/O Receive Control and Status Register	RE	IS	RW1	RW0	RC	SIS	RESERVED	
P10		$FC14	Serial I/O Transmit Control and Status Register	TE	AT	LM	TW1	TW0	TC	P/S	RES
P11		$FC16	Timer B Latch								
P12		$FC18	Timer A Low Latch								
P13		$FC1A	Timer A High Latch								
P14		$FC1C	Timer and Handshake Control Register	HSM2	HSM1	HSM0	XI2C	HSE	TCE	TCOC	TAM1
P15		$FC1E	Expanded Bus Control and Status Register	SEG1	SEG0	BLCK	F/S	EXP	F/P	TEST 1*	TEST 0*
P16		$FC20	Port 0 Direction Control (DDR0)								
P17		$FC22	Port 1 Direction Control (DDR1)								
P18		$FC24	Serial I/O Mode and Sync Register	A/S	WL1	WL0	ST	PAR1	PAR0	TCO	WS
P19		$FC26	Timer C Latch								

***NOTE:** Both test bits should remain a 0 during normal operation.

Port Map

7	6	5	4	3	2	1	0	Initial Condition
								—
I/O	**XI2**	**XI1**	**XI0**	**SI**	**RCLK**	**TCLK**	**SO**	—
← RESERVED →								—
TRANSMIT BUFFER								—
← RESERVED →								High Byte 00-----00
← RESERVED →								—
← RESERVED →								—
RSCI	RNI	XI1I	TBOI	TBII	XI0I	XMTI	TCI	$0000
RSCM	RNM	XI1M	TBOM	TBIM	XI0M	XMTM	TCM	$0000
BF	OE	PE	FE	SF/AF	← RESERVED →			$0000
BE	UE	END	← RESERVED →					$00A0
								—
								—
								—
TAM0	TAE	TAIC	TAOC	TBM1	TBM0	TBE	TBIC	$0000
UPC/GP	R/G	← RESERVED →						Mode pin: VCC-$0400 CLKOUT-$0C00 GND-$0800
								$0000
								$0000
SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	$0000
								—

NOTE:
When a reserved bit is read, it is read as a zero.

ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin relative to GND . . . . . −1.0 V to +7.0 V
Total Device Power Dissipation . . . . . 1 Watt
Ambient Storage Temperature . . . . . −65°C to +150°C
Ambient Operating Temperature (T_L<T_A<T_H) . . . . .
Commercial MK68HC2xx/xxx-Cx . . . . . −0°C to +70°C
Industrial MK68HC2xx/xxx-Vx . . . . . −40°C to +85°C
Military MK68HC2xx/xxx-Mx . . . . . −55°C to +125°C

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

MK68200 DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 V ± 10%, T_A = T_L to T_H)

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
V_{IL}	Input low voltage; all inputs	−0.3	0.8	V	
V_{IH}	Input high voltage; all inputs except CLK1	2.0	V_{CC} + 0.3	V	
V_{IH}	Input high voltage; CLK1	0.7 × V_{CC}	V_{CC} + 0.3	V	
V_{OL}	Output low voltage; all outputs		0.4	V	I_{OL} = 2.0 mA
V_{OH}	Output high voltage; all outputs	2.4		V	I_{OH} = −250μA
V_{OH}	Output high voltage; all outputs	V_{CC} −0.1		V	I_{OH} = −10μA
I_{LI}	Input leakage current		±1	υA	V_{IN} = 0 to V_{CC}
I_{LO}	Three-state output leakage current in float		±10	μA	V_{OUT} = 0.4 V to V_{CC}
I_{CC1}	Input power supply current CLKOUT = 12.5 MHz CLKOUT = 10 MHz CLKOUT = 8 MHz CLKOUT = 0.5 MHz		 75 65 55 15	 mA mA mA mA	Outputs Open
I_{CC2}	Power supply current in IDLE mode. CLKOUT = 12.5 MHz CLKOUT = 10 MHz CLKOUT = 8 MHz CLKOUT = 0.5 MHz		 20 20 20 10	 mA mA mA mA	Outputs Open. All I/O functions active.
I_{CC3}	Power supply current in STOP mode		5	mA	Outputs Open

CAPACITANCE

T_A = 25°C

SYMBOL	PARAMETER	MAX	UNIT	TEST CONDITION
C_{IN}	Input Capacitance	10	pF	Unmeasured pins returned to ground
C_{OUT}	Three-state Output Capacitance	10	pF	

MK68HC200 AC ELECTRICAL SPECIFICATIONS

$T_A = T_L$ to T_H, V_{CC} = +5V ±10% unless otherwise specified.
AC measurements are referenced from minimum V_{IH} or maximum V_{IL} for inputs and from minimum V_{OH} or maximum V_{OL} for outputs.

		8 MHz		10 MHz		12.5 MHz			
NO.	DESCRIPTION	MIN	MAX	MIN	MAX	MIN	MAX	UNITS	NOTES
1	$\overline{\text{RESET}}$ low time	20		20		20		state times	1
2	CLK 1 width high (external clock input)	22		18		14		ns	
3	CLK 1 width low (external clock input)	22		18		14		ns	
4	CLK 1 period (external clock input)	62	1000	50	1000	40	1000	ns	
5	Crystal input frequency	1.000	16.000	1.000	20.000	1.000	25.000	MHz	
6	Clock Period (PHI 1) (tc)	125		100		80		ns	
7	PHI 1 low to PHI 1 high	62		50		40		ns	
8	PHI 1 high to PHI 1 low	62		50		40		ns	
9	PHI 1 low to CLKOUT low		20		16		13	ns	
10	PHI 1 high to CLKOUT high		20		16		13	ns	

MK68HC200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (UPC, GP, AND PRIVATE BUSES)

		8 MHz		10 MHz		12.5 MHz			
NO.	DESCRIPTION	MIN	MAX	MIN	MAX	MIN	MAX	UNITS	NOTES
11	PHI 1 low to R/$\overline{\text{W}}$, $\overline{\text{HB}}$, $\overline{\text{LB}}$, FC1, FC2, $\overline{\text{EXPMC}}$, $\overline{\text{DMODE}}$ Valid		58		46		37	ns	2
12	PHI 1 high to $\overline{\text{AS}}$ low		58		46		37	ns	2
13	PHI 1 low to address valid		60		50		40	ns	2
14	$\overline{\text{AS}}$ low to address invalid	35		30		24		ns	2
15	PHI 1 low to tri-state address		45		36		29	ns	2
16	Tri-state address to $\overline{\text{DS}}$, $\overline{\text{LDS}}$, or $\overline{\text{UDS}}$ starting low (fast cycle)	10		10		10		ns	2
17	PHI 1 low to $\overline{\text{DS}}$, $\overline{\text{LDS}}$, or $\overline{\text{UDS}}$ low (fast cycle)		83		66		53	ns	2
18	PHI 1 low to data out valid during write		60		50		40	ns	2
19	PHI 1 low to R/$\overline{\text{W}}$, $\overline{\text{HB}}$, $\overline{\text{LB}}$, FC1, FC2, $\overline{\text{EXPMC}}$, $\overline{\text{DMODE}}$ invalid	0		0		0		ns	2
20	PHI 1 low to address/data bus driven	0		0		0		ns	2
21	$\overline{\text{AS}}$ low to $\overline{\text{DS}}$, $\overline{\text{LDS}}$, or $\overline{\text{UDS}}$ starting low (fast cycle)	50	113	42	90	33	72	ns	2
85	$\overline{\text{AS}}$ high to FPRIV invalid	0		0		0		ns	
86	Address valid to FPRIV valid (fast cycle)		121		94		75	ns	
87	Valid data setup to $\overline{\text{DS}}$, $\overline{\text{UDS}}$ or $\overline{\text{LDS}}$ high during write (fast cycle)	65		50		40		ns	2

MK68HC200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (UPC AND GP BUSES)

NO.	DESCRIPTION	8 MHz		10 MHz		12.5 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX	MIN	MAX		
22	Tri-state address to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ starting low (standard cycle)	67		54		43		ns	
23	PHI 1 high to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ low (standard cycle)		83		66		53	ns	2
24	Valid Data Setup to PHI 1 low	5		5		5		ns	2
25	$\overline{AS}$ low to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ starting low (standard cycle)	112	175	90	138	72	111	ns	2
26	R/$\overline{W}$, $\overline{HB}$, or $\overline{LB}$ valid to $\overline{AS}$ starting low	45		36		29		ns	
27	Address valid to $\overline{AS}$ starting low	45		36		29		ns	
28	Input data hold time from PHI 1 low	22		18		14		ns	
29	Input data hold time from $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ high	0		0		0		ns	
30	PHI 1 low to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ high		90		72		58	ns	
31	$\overline{DTACK}$ low or FPRIV valid setup to PHI 1 high	7		6		5		ns	
32	$\overline{LDS}$, $\overline{UDS}$, or $\overline{DS}$ high to $\overline{DTACK}$ high (hold time)	0		0		0		ns	
33	$\overline{LDS}$, $\overline{UDS}$, or $\overline{DS}$ pulse width	120		90		72		ns	
34	PHI 1 high to $\overline{AS}$ high		45		36		29	ns	
35	PHI 1 low to data out invalid	0		0		0		ns	
36	$\overline{AS}$ inactive	117		90		72		ns	
37	$\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ high to data out invalid	90		66		53		ns	
38	$\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ high to $\overline{AS}$ high	5		5		5		ns	
88	Address valid to FPRIV valid (standard cycle)		246		194		155	ns	
89	Valid data setup to $\overline{DS}$, $\overline{LDS}$, or $\overline{UDS}$ during write (standard cycle)	190		150		120		ns	2
90	$\overline{DS}$, $\overline{LDS}$ or $\overline{UDS}$ low to data in valid (fast cycle)		37		29		22	ns	2
91	$\overline{DS}$, $\overline{LDS}$ or $\overline{UDS}$ low to data in valid (standard cycle)		100		79		62	ns	2
92	Address valid to data in valid (fast cycle)		185		145		115	ns	
93	Address valid to data in valid (standard cycle)		310		245		195	ns	

MK68HC200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (UPC BUS)

NO.	DESCRIPTION	8 MHz		10 MHz		12.5 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX	MIN	MAX		
39	$\overline{BGACK}$ low to $\overline{BR}$ high	100	225	75	180	60	144	ns	
40	$\overline{BG}$ low to $\overline{BGACK}$ low	50	300	50	240	40	192	ns	
41	$\overline{BGACK}$, $\overline{AS}$, $\overline{DTACK}$, inactive to $\overline{BGACK}$ low; $\overline{BG}$ already low	0	300	0	240	0	192	ns	
42	$\overline{BGACK}$ low to $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$ or address/data bus driven	40	68	30	54	25	43	ns	
43	$\overline{AS}$, $\overline{LDS}$, $\overline{UDS}$, R/$\overline{W}$ or address/data bus tri-state to $\overline{BGACK}$ high	0	90	0	72	0	58	ns	

MK68HC200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (GP BUS)

NO.	DESCRIPTION	8 MHz		10 MHz		12.5 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX	MIN	MAX		
45	$\overline{BUSIN}$ low to $\overline{BUSOUT}$ low (bus grantor, fast cycle, no wait states)		950		720		576	ns	
46	$\overline{BUSOUT}$ high to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$ or address/data bus driven (bus grantor)	15		15		15		ns	
47	$\overline{BUSIN}$ high to $\overline{BUSOUT}$ high (bus grantor)	260	450	180	360	144	288	ns	
48	Tri-state $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$ or address /data bus to $\overline{BUSOUT}$ low (bus grantor)	50		42		33		ns	
50	$\overline{BUSOUT}$ low to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$ or address/data bus driven (bus requestor, $\overline{BUSIN}$ low)	120		90		72		ns	
51	$\overline{BUSIN}$ low to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{LB}$, $\overline{HB}$ or address/data bus driven (bus requestor, $\overline{BUSOUT}$ low)	135	325	108	300	86	240	ns	
53	$\overline{BUSOUT}$ high to $\overline{BUSIN}$ high (bus requestor)		275		240		192	ns	
55	Tri-state $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{HB}$, $\overline{LB}$ or address/data bus to $\overline{BUSOUT}$ high (bus requestor)	50		39		31		ns	

MK68HC200 BUS AC ELECTRICAL SPECIFICATIONS (PRIVATE BUS)

NO.	DESCRIPTION	8 MHz		10 MHz		12.5 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX	MIN	MAX		
56	Valid Data Setup to PHI 1 low	15		12		10		ns	
57	PBR/$\overline{W}$ valid to $\overline{PBAS}$ starting low	30		24		19		ns	
58	Address valid to $\overline{PBAS}$ starting low	27		21		17		ns	
59	Input data hold time from PHI 1 low	0		0		0		ns	
60	Input data hold time from $\overline{PBDS}$ high	0		0		0		ns	

MK68HC200 EXPANDED BUS AC ELECTRICAL SPECIFICATIONS (PRIVATE BUS) (Cont.)

NO.	DESCRIPTION	8 MHz		10 MHz		12.5 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX	MIN	MAX		
61	PHI 1 low to $\overline{\text{PBDS}}$ high		80		63		50	ns	
62	$\overline{\text{PBDTACK}}$ low setup to PHI 1 high	10		9		7		ns	
63	$\overline{\text{PBDS}}$ high to $\overline{\text{PBDTACK}}$ high (hold time)	−15		−15		−15		ns	
64	$\overline{\text{PBDS}}$ pulse width	95		75		60		ns	
65	PHI 1 high to $\overline{\text{PBAS}}$ high		58		45		36	ns	
66	PHI 1 low to data out invalid	10		10		10		ns	
67	$\overline{\text{PBAS}}$ inactive	100		81		65		ns	
68	$\overline{\text{PBDS}}$ high to data out invalid	100		81		65		ns	
69	$\overline{\text{PBDS}}$ high to $\overline{\text{PBAS}}$ high	15		15		15		ns	
94	$\overline{\text{PBDS}}$ low to data in valid		27		27		37	ns	
95	Address valid to data in valid		159		159		238	ns	

MK68HC200 INPUT/OUTPUT AC ELECTRICAL CHARACTERISTICS

NO.	DESCRIPTION		8 MHz		10 MHz		12.5 MHz		UNITS	NOTES
			MIN	MAX	MIN	MAX	MIN	MAX		
70	Active and inactive pulse times	For XI2, XI1, STRH, STRL, TAI, TBI, NMI	5		5		5		state times	1
		For XI0	3		3		3			
71	Input data setup to falling edge of STRH, STRL		7		6		5		ns	
72	Input data hold from the falling edge of STRH, STRL		30		24		19		ns	
73	RDYH, RDYL low time		1	3	1	3	1	3	state times	1
74	Delay from STRH, STRL high to RDYH, RDYL low			55		45		36	ns	
75	Delay from data valid to RDYH, RDYL high (output mode)			3		3		3	state times	1
76	Delay from STRH high to data out (bidirectional mode)			45		36		29	ns	
77	Port 0 data hold time from STRH low (bidirectional mode)		15		12		10		ns	
78	Delay to Port 0 float from STRH low (bidirectional mode)			43		33		26	ns	
79	TCLK,RCLK period (asynchronous)	as input	.125	DC	.100	DC	.080	DC	μs	
		as output	.250	DC	.200	DC	.160	DC		
	TCLK,RCLK period (synchronous)		.500	DC	.400	DC	.320	DC		
80	TCLK, RCLK width low or high (asynchronous)	as input	1	DC	1	DC	1	DC	state times	1
		as output	2	DC	2	DC	2	DC		
	TCLK, RCLK width low or high (synchronous)	as input	3	DC	3	DC	3	DC	state times	1
		as output	4	DC	4	DC	4	DC		

MK68HC200 INPUT/OUTPUT AC ELECTRICAL SPECIFICATIONS

NO.	DESCRIPTION		8 MHz MIN	8 MHz MAX	10 MHz MIN	10 MHz MAX	12.5 MHz MIN	12.5 MHz MAX	UNITS	NOTES
82	TCLK low to SO delay (sync mode)	TCLK as input	165		132		106		ns	
		TCLK as output	37		30		24			
83	SI to RCLK high setup time (sync mode)	RCLK as input	15		12		10		ns	
		RCLK as output	90		72		58			
84	SI hold time from RCLK high (sync mode)	RCLK as input	22		18		14		ns	
		RCLK as output	0		0		0			

NOTES
1. One state time is equal to one-half of the instruction clock (PHI 1) period.
2. For the private bus case, the signals referenced apply to the equivalent private bus signals.

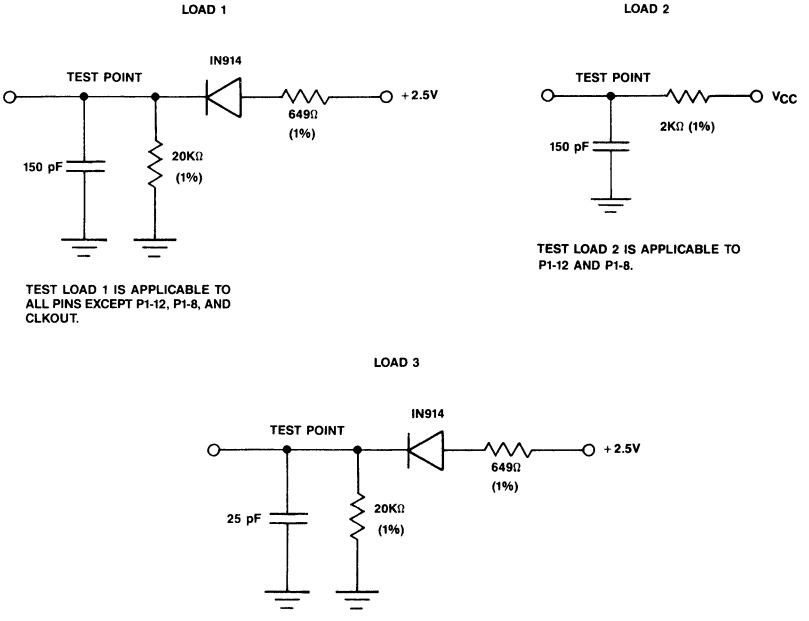


Figure 18. Output Test Load

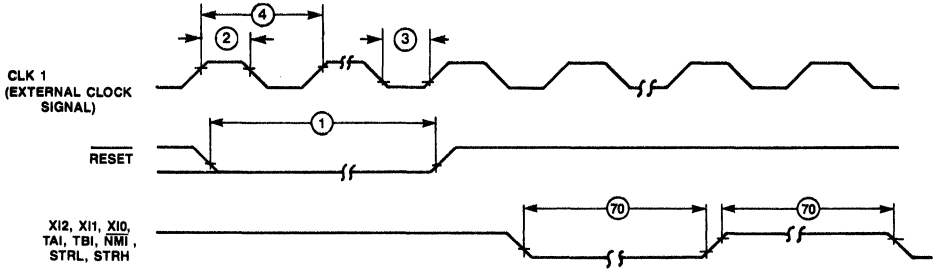


Figure 19. MK68HC200 AC Timing

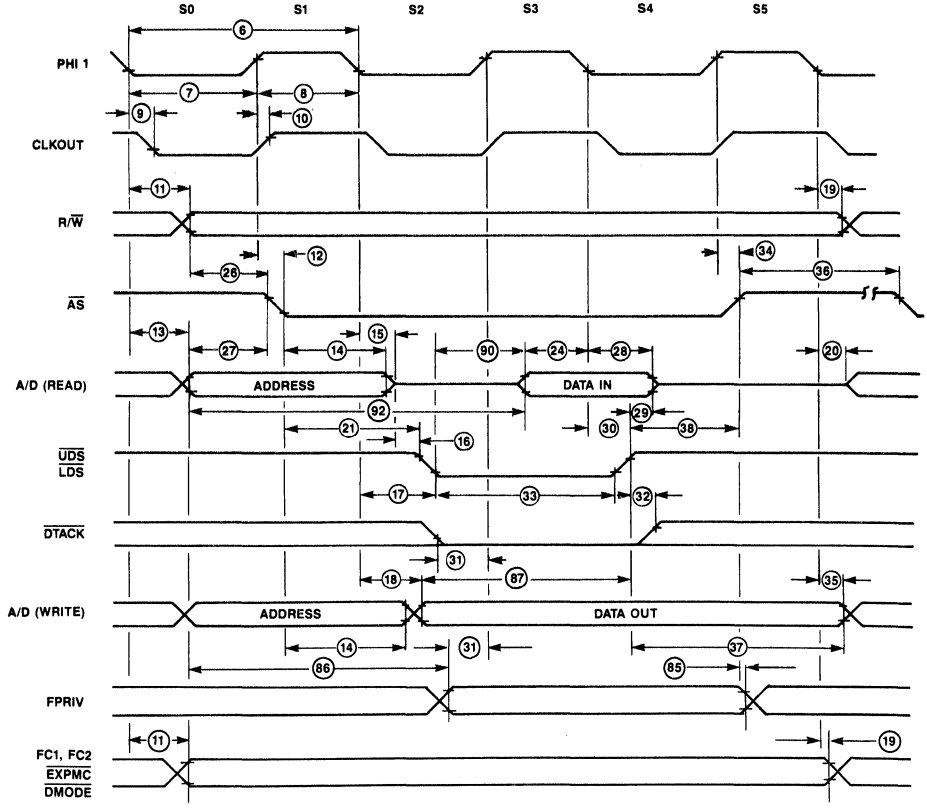


Figure 20. MK68HC201 UPC Bus Timing (Fast Cycle)

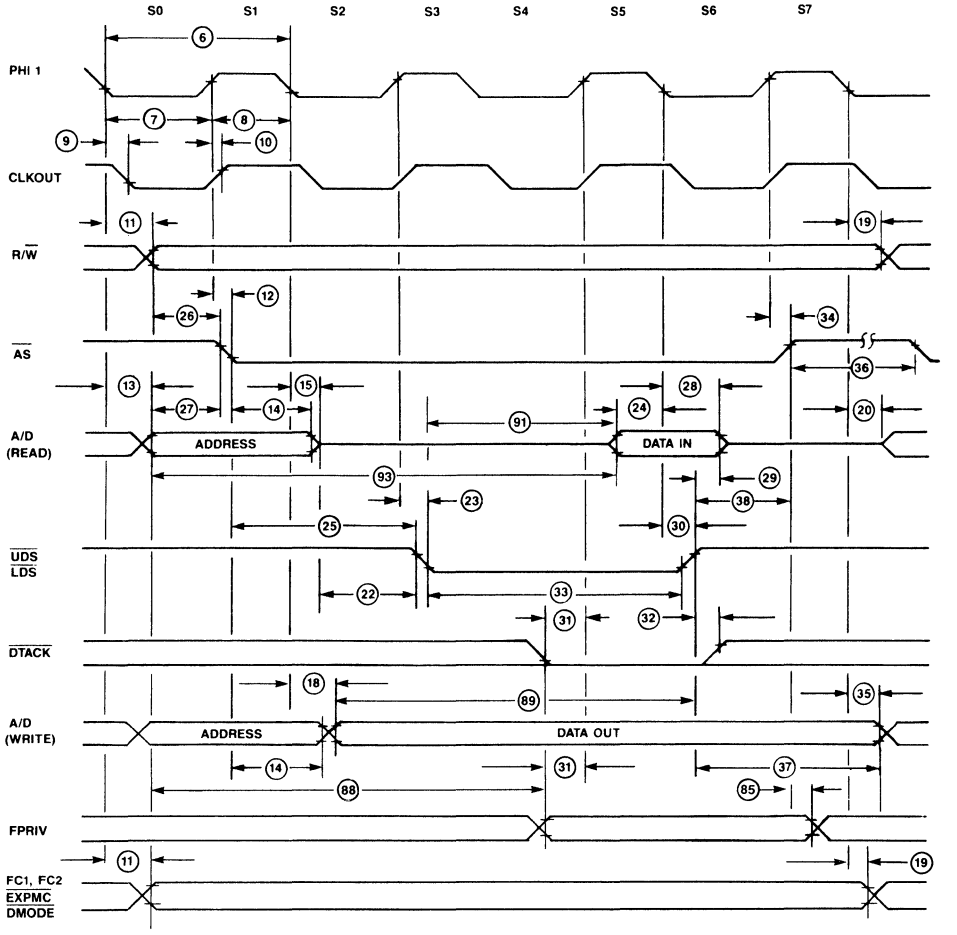


Figure 21. MK68HC201 UPC Bus Timing (Standard Cycle)

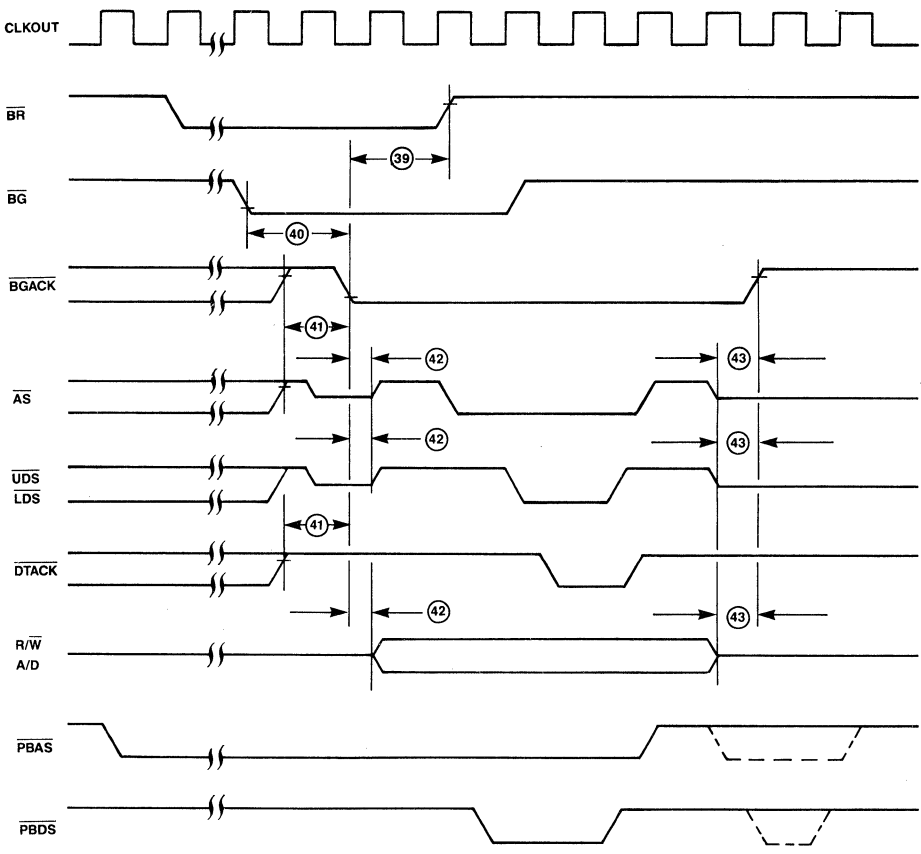


Figure 22. MK68HC201 UPC Bus Arbitration Timing

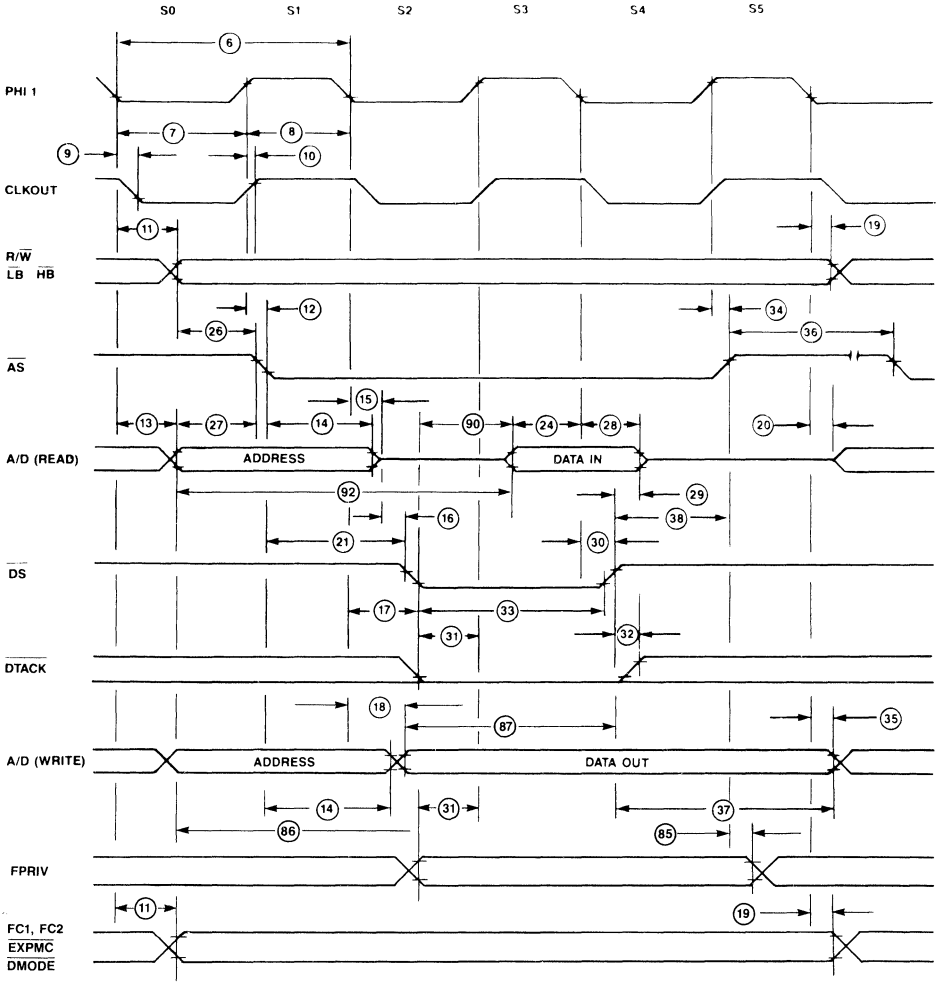


Figure 23. MK68HC211 GP Bus Timing (Fast Cycle)

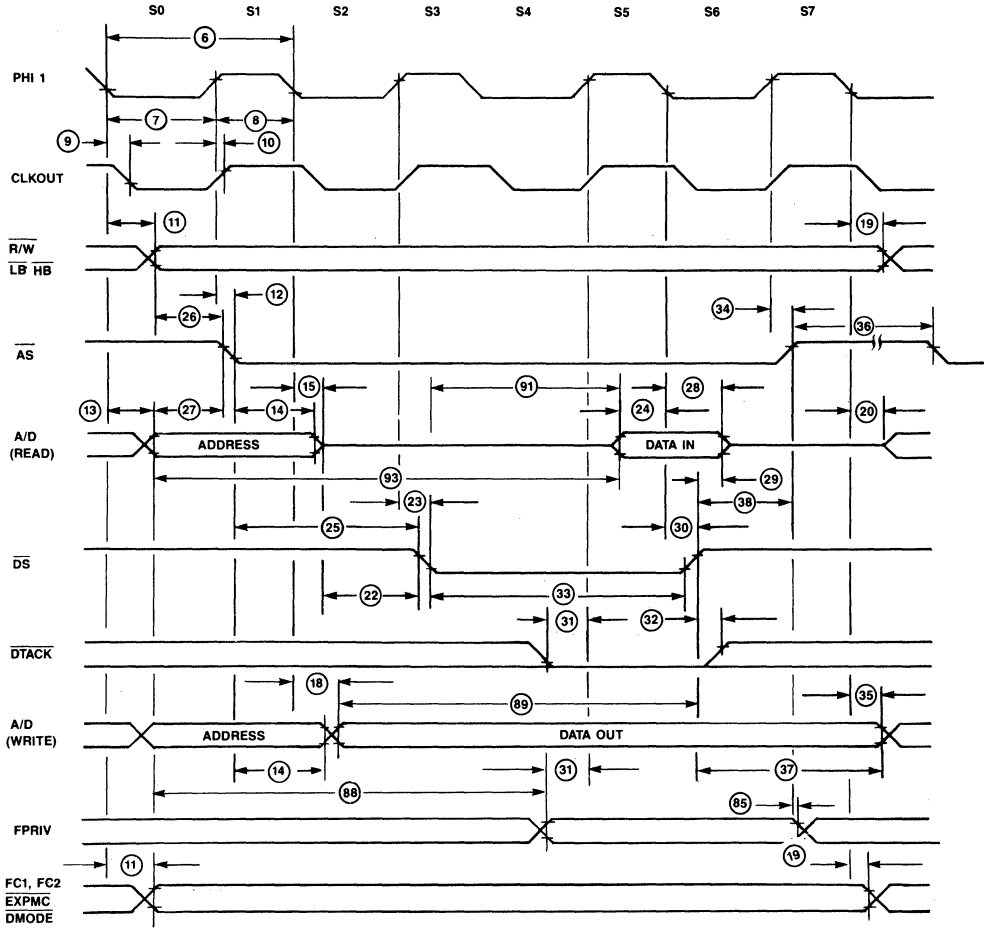


Figure 24. MK68HC211 GP Bus Timing (Standard Cycle)

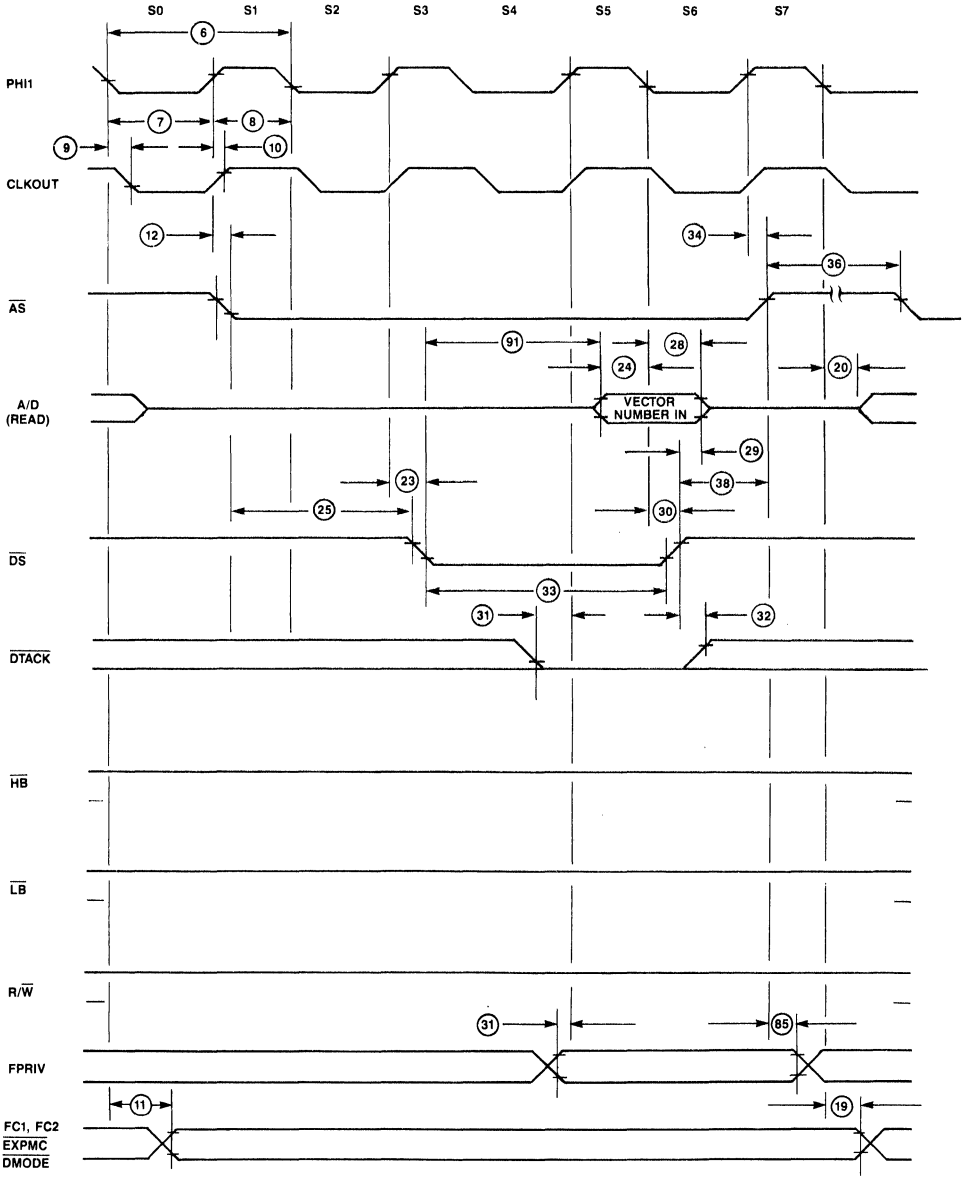


Figure 25. MK68HC211 GP Bus Timing (Interrupt Acknowledge Timing)

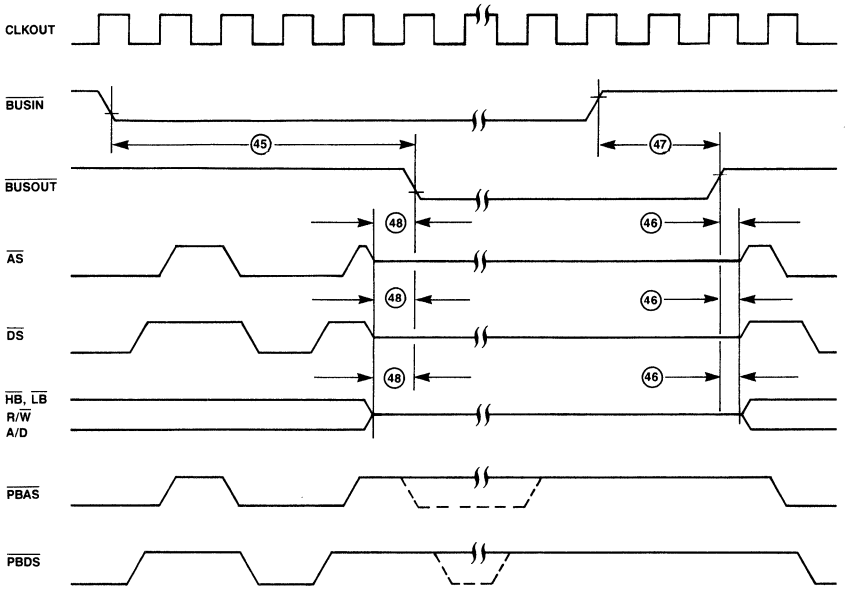


Figure 26. MK68HC211 GP Bus Arbitration Timing (Bus Grantor)

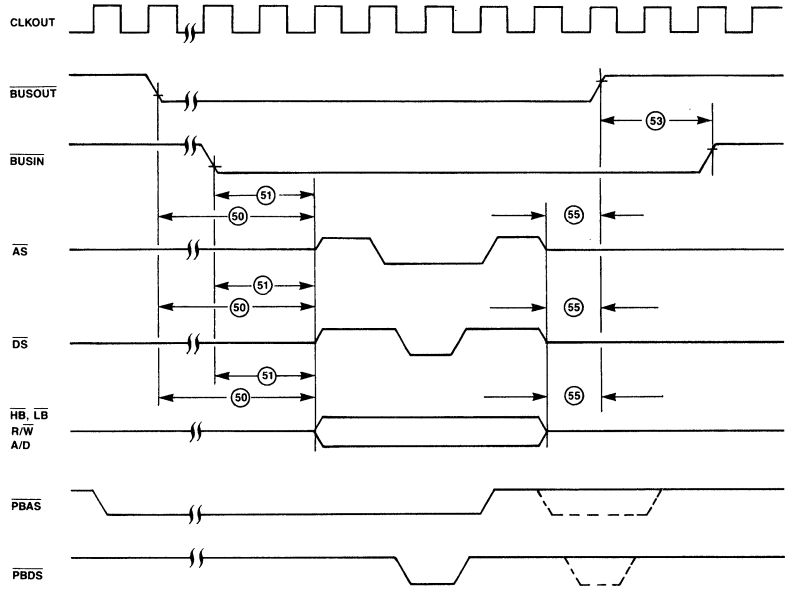


Figure 27. MK68HC211 GP Bus Arbitration Timing (Bus Requestor)

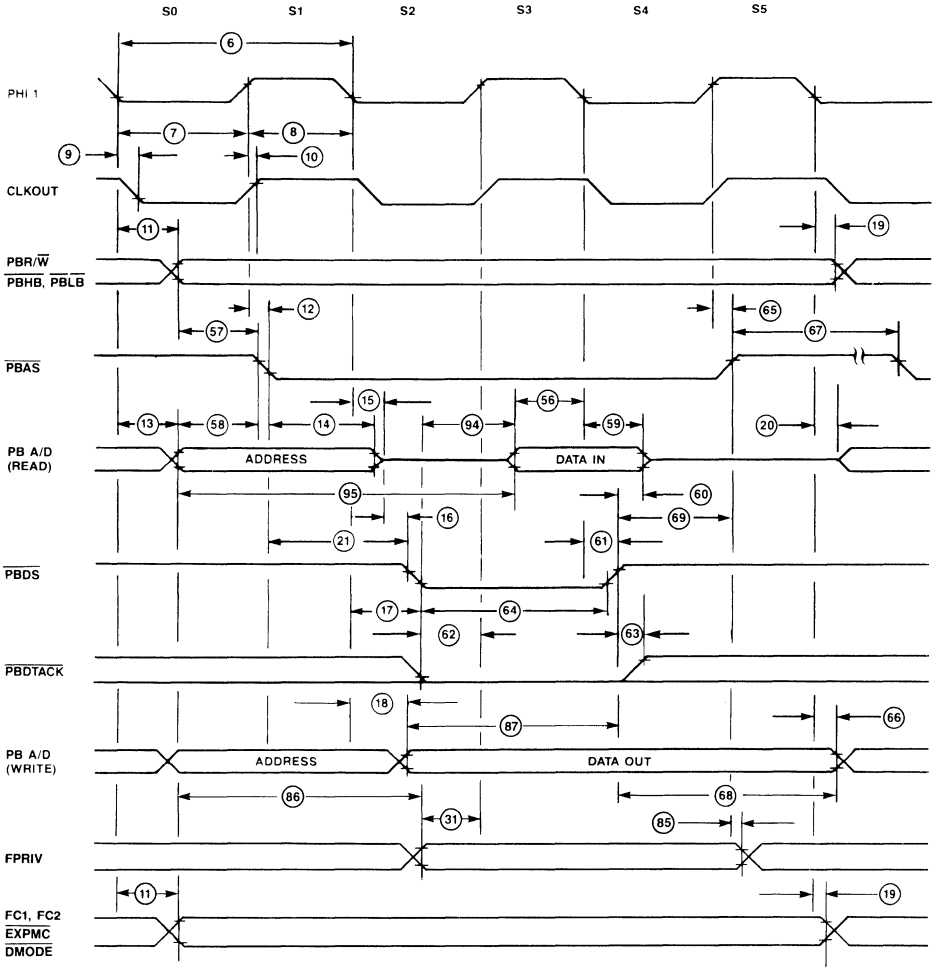


Figure 28. MK68HC221 Private Bus Timing (Fast Cycle)

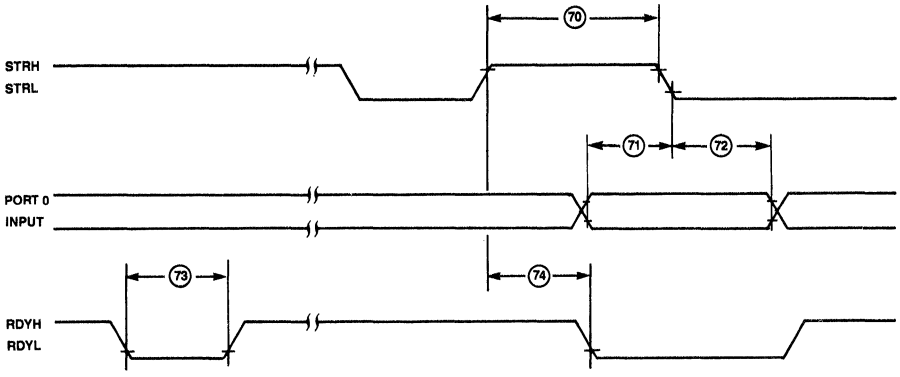


Figure 29. Input/Output AC Timing (Data Input)

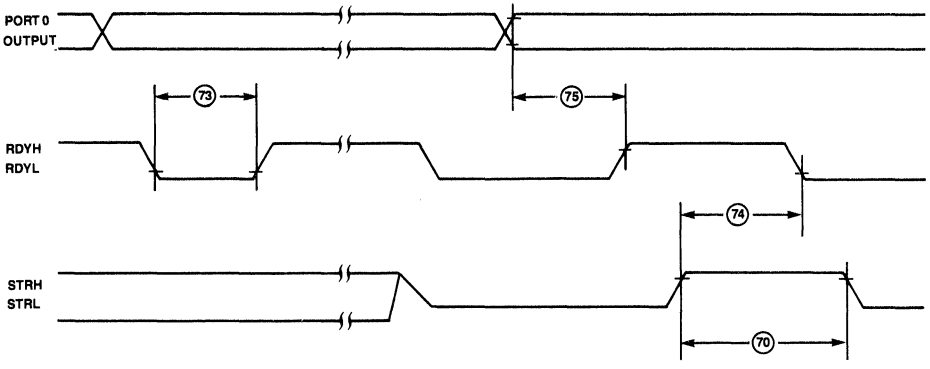


Figure 30. Input/Output AC Timing (Data Output)

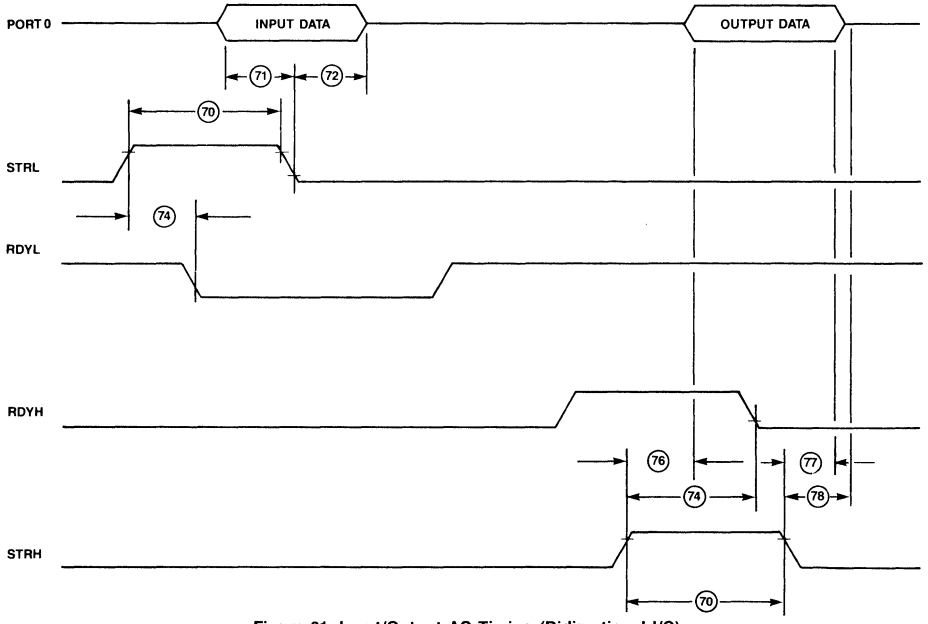


Figure 31. Input/Output AC Timing (Bidirectional I/O)

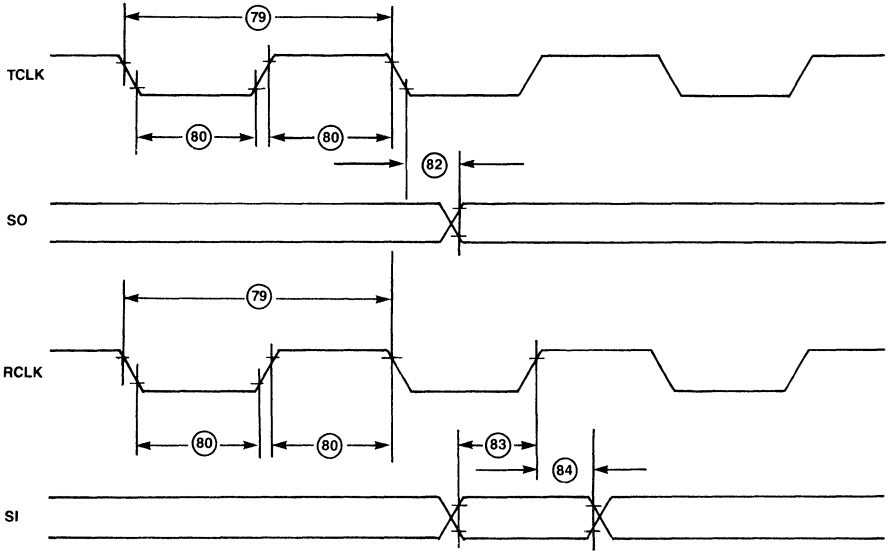


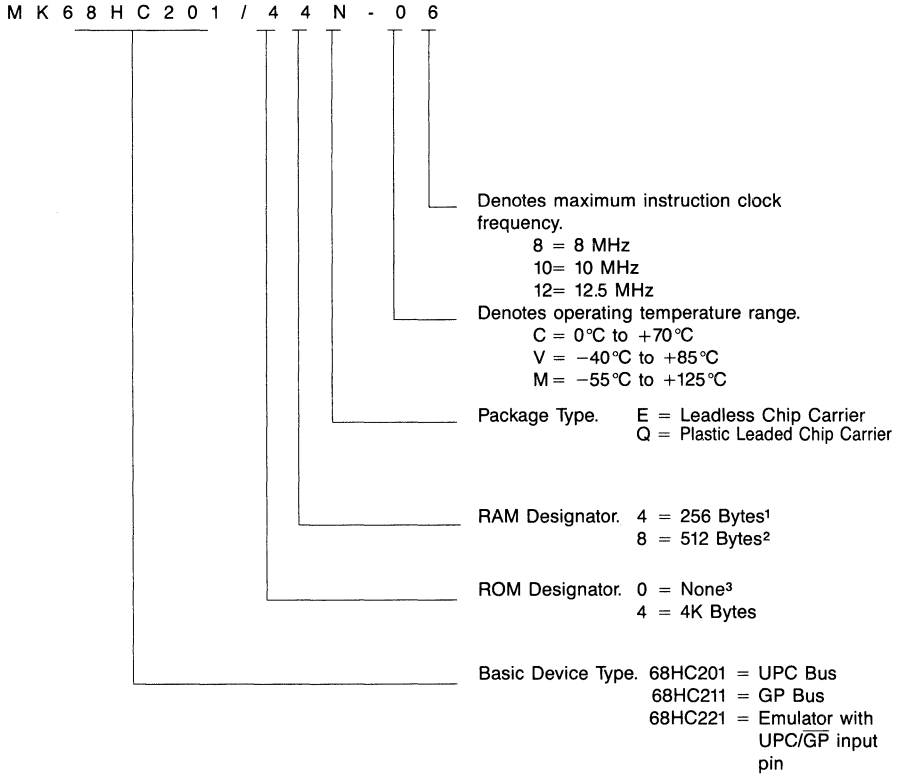
Figure 32. Input/Output AC Timing (Serial I/O)

PART NUMBERING INFORMATION

There are two types of part numbers for the MK68HC200 family of devices. The generic part number describes the basic device type, the amount of ROM and RAM, the desired package type, temperature range, power supply tolerance, and expandable bus interface type. The device order number indicates the specific mask set Mostek will use to manufacture the device, along with package type, speed grade and temperature range.

Generic Part Number

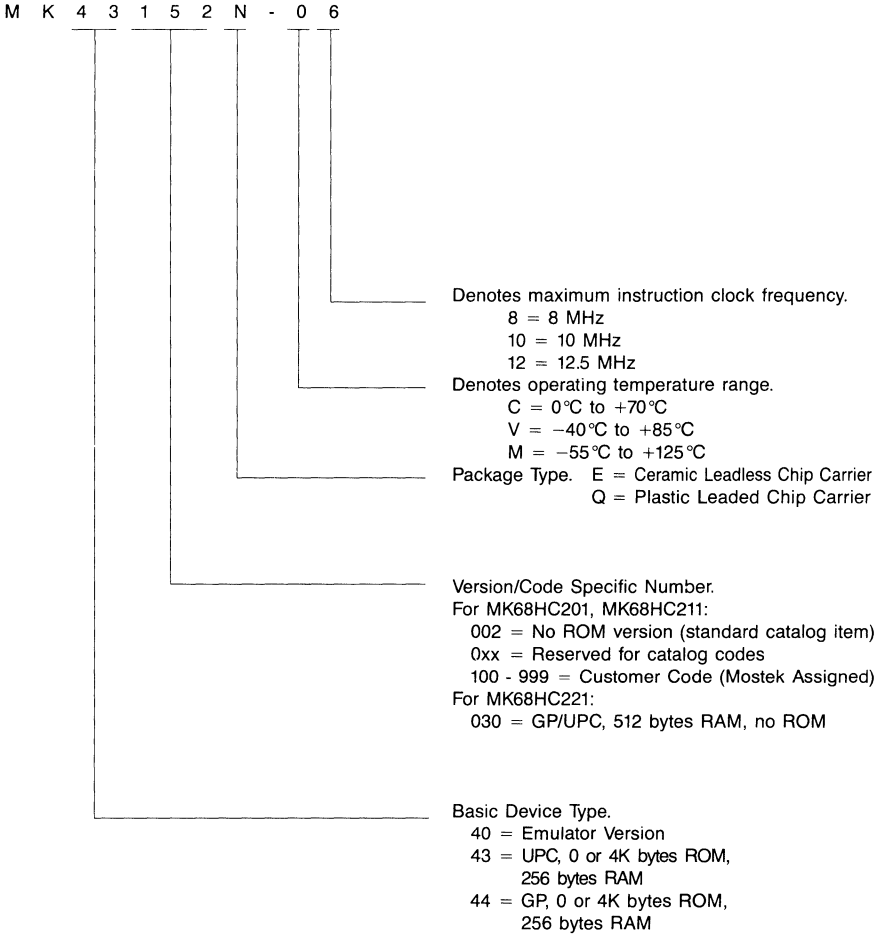
An example of the generic part number is shown below:



NOTES
1. Must be "4" when specifying the ROM version.
2. Must be "8" when specifying the emulator version.
3. Must be "0" when specifying the emulator version.

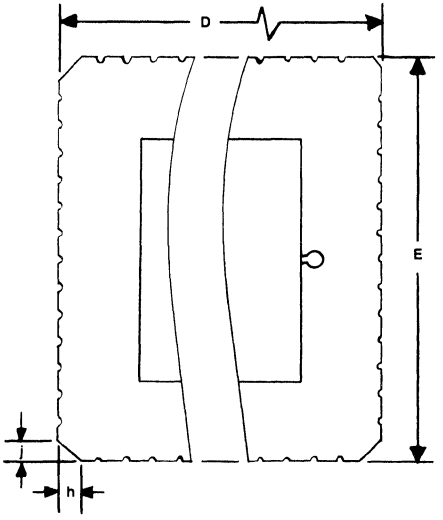
Device Order Number

An example of the device order number is shown below:

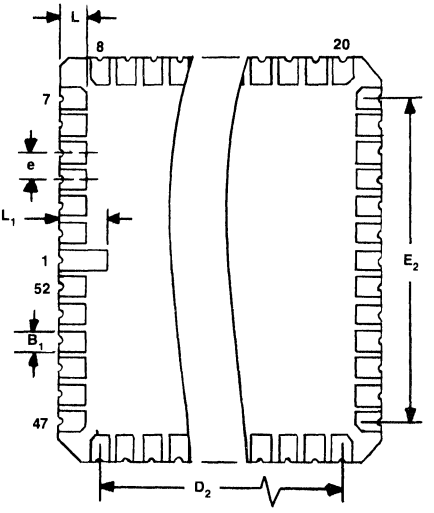


PART NUMBER EXAMPLES (A noninclusive list)	
MK68HC201/44Q-C10	Device Order Number = MK43XXXQ-C10 Speed = 10MHz Temperature = 0° to 70°C Package = 52 pin plastic LCC RAM = 256 bytes ROM = 4096 bytes Bus = UPC
MK68HC211/04Q-C8	Device Order Number = MK44002Q-C8 Speed = 8MHz Temperature = 0° to 70°C Package = 52 pin plastic LCC RAM = 256 bytes ROM = None Bus = GP
MK68HC221/08E-M12	Device Order Number = MK40030E-M12 Speed = 12.5MHz Temperature = −55°C to +125°C Package = 84 pin ceramic LCC RAM = 512 bytes ROM = None Bus = GP/UPC
MK68HC201/44Q-V8	Device Order Number = MK43XXXQ-V8 Speed = 8MHz Temperature = −40 to +85°C Package = 52 pin plastic LCC RAM = 256 bytes ROM = 4096 bytes Bus = UPC

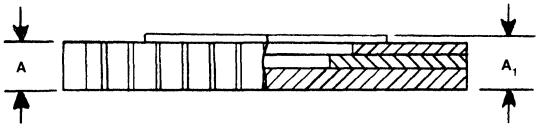
MK68HC200 52-Pin
Ceramic Leadless Chip Carrier (E)



TOP VIEW



BOTTOM VIEW

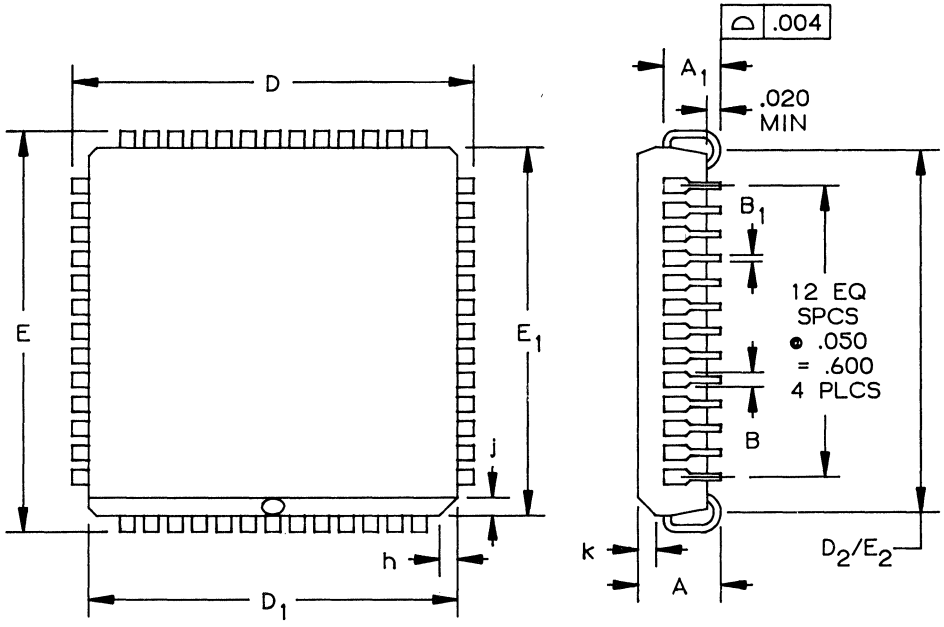


DIM.	INCHES MIN.	INCHES MAX.	NOTES
A	.070	.095	
A_1	.080	.110	
B_1	.022	.028	
D	.739	.761	
D_2	.590	.610	
E	.739	.761	
E_2	.590	.610	
e	.048	.052	
h	.035	.045	
j	.035	.045	
L	.045	.055	
L_1	.075	.095	

1 INCH = 2.54 CENTIMETERS

NOTES:
1. Body material shall be ceramic.
2. Plating shall be gold over nickel as specified in the detail specification.

MK68HC200 52-Pin
Plastic Leaded Chip Carrier (Q)



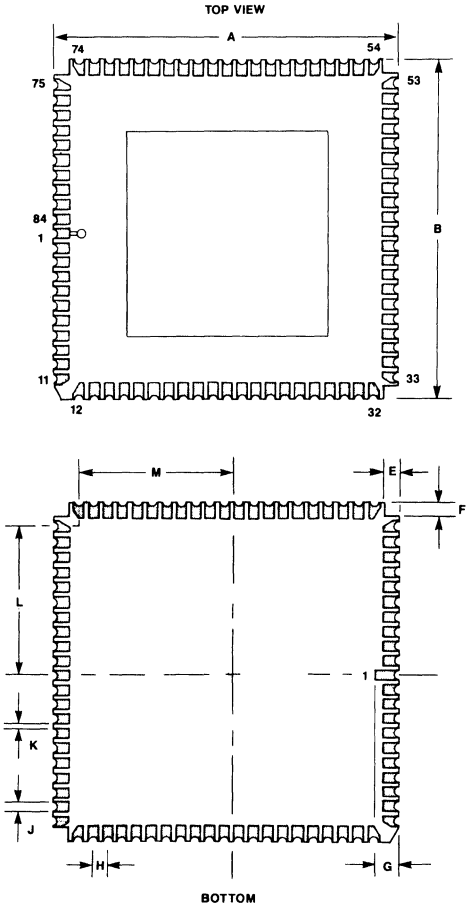
DIM.	INCHES		NOTES
	MIN.	MAX.	
A	.165	.185	2
A_1	.090	.120	2
B	.026	.032	2
B_1	.013	.021	2
D	.785	.795	
D_1	.750	.756	
D_2	.690	.730	
E	.785	.795	
E_1	.750	.756	
E_2	.690	.730	
h	.042	.060	
j	.042	.060	
k	.042	.056	

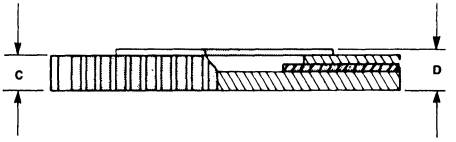
1 INCH = 2.54 CENTIMETERS

NOTES:

1. LEAD FINISH TO BE SPECIFIED PER CUSTOMER AGREEMENT.
2. WHEN SOLDER DIP LEAD FINISH IS SPECIFIED, THE MAXIMUM LIMIT SHALL BE INCREASED BY .003.

MK68HC221 84-Pin
Ceramic Leadless Chip Carrier (E)

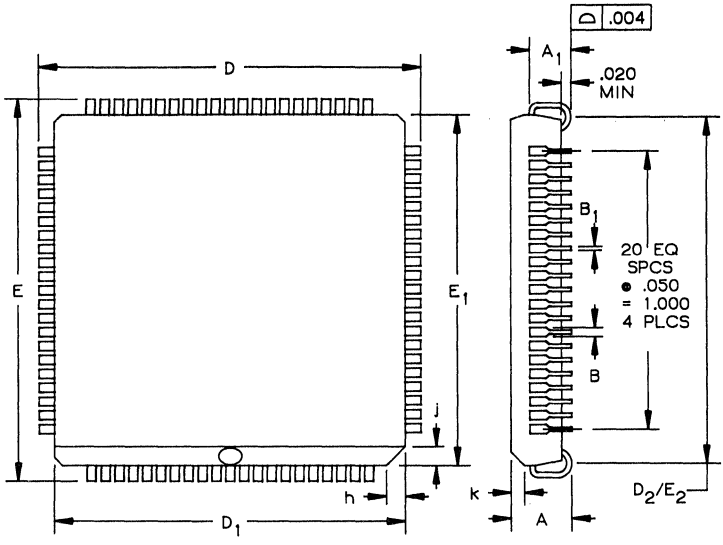




DIM.	INCHES	
	MIN	MAX
A	1.138	1.167
B	1.138	1.167
C	0.070	0.090
D	0.080	0.110
E	0.044	0.056
F	0.044	0.056
G	0.075	0.095
H	0.048	0.052
J	0.033	0.039
K	0.010	0.018
L	0.495	0.505
M	0.495	0.505

1 INCH = 2.54 CENTIMETERS

MK68HC221 84-Pin
Plastic Leaded Chip Carrier (Q)



DIM.	INCHES		NOTES
	MIN.	MAX.	
A	.165	.200	2
A_1	.090	.130	2
B	.026	.032	2
B_1	.013	.021	2
D	1.185	1.195	
D_1	1.150	1.158	
D_2	1.090	1.130	
E	1.185	1.195	
E_1	1.150	1.158	
E_2	1.090	1.130	
h	.042	.060	
j	.042	.060	
k	.042	.056	

1 INCH = 2.54 CENTIMETERS

NOTES:

1. LEAD FINISH TO BE SPECIFIED PER CUSTOMER AGREEMENT.
2. WHEN SOLDER DIP LEAD FINISH IS SPECIFIED, THE MAXIMUM LIMIT SHALL BE INCREASED BY .003.

CHAPTER 4 - 68000 PERIPHERALS

68000 PERIPHERALS SELECTION GUIDE

Part number	Description	Technology	Alt source	CLK freq. (MHz)	Page
MK68230-8 MK68230-10	Parallel Interface Timer (PI/T) - 68000 bus compatible 24-bit programmable timer modes	HMOS	MC68320-8 MC68230-10	8 10	4-3
MK68451-8 MK68451-10 MK68451-12	Memory management unit - Compatible with TS68000 and TS68008	NMOS	MC68451-8 MC68451-10	8 10	4-13
MK68564-04 MK68564-05	Dual serial Input Output controller asynchronous, synchronous byte-oriented and synchronous bit oriented protocols	HMOS	— —	4 5	4-17
MK68901-00 MK68901-05	Multifunction peripheral - 8 I/O - 16 Source interrupt controller - Single channel USART full duplex 68000 bus compatible	HMOS	—	4 5	4-29
TS68HC901	CMOS version of the MK68901	HCMOS		4, 5, 8	4-59

THOMSON COMPONENTS MOSTEK

MK68230

PARALLEL INTERFACE/TIMER (PI/T)

PRELIMINARY

MICROCOMPUTER COMPONENTS

FEATURES

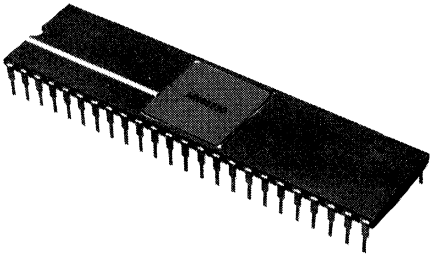
- ☐ 68000 Bus Compatible
- ☐ Port Modes Include:
  - Bit I/O
  - Unidirectional 8-bit and 16-bit
  - Bidirectional 8-bit and 16-bit
- ☐ Programmable Handshaking Options
- ☐ 24-bit Programmable Timer Modes
- ☐ Five Separate Interrupt Vectors
- ☐ Separate Port and Timer Interrupt Service Requests
- ☐ Registers are Read/Write and Directly Addressable
- ☐ Registers are Addressed for MOVEP (Move Peripheral) and DMAC Compatibility

GENERAL DESCRIPTION

The MK68230 Parallel Interface/Timer (PI/T) provides versatile double-buffered parallel interfaces and an operating system oriented timer to MK68000 systems. The parallel interfaces operate in unidirectional or bidirectional modes, either 8 or 16 bits wide. In the unidirectional modes, an associated data direction register determines whether the port pins are inputs or outputs. In the bidirectional modes, the data direction registers are ignored, and the direction is determined dynamically by the state of four handshake pins. These programmable handshake pins provide an interface flexible enough for connection to a wide variety of low, medium, or high speed peripherals or other computer systems. The PI/T ports allow use of vectored or autovectored interrupts, and also provide a DMA request pin for connection to the Direct Memory Access Controller or a similar circuit. The PI/T timer contains a 24-bit wide counter and a 5-bit prescaler. The timer may be clocked by the system clock (PI/T CLK pin) or by an external clock (TIN pin), with the option of using a 5-bit prescaler. It can generate periodic interrupts, a square wave, or a single interrupt after a programmed timer period. Also, it can be used for elapsed time measurement or as a device watchdog.

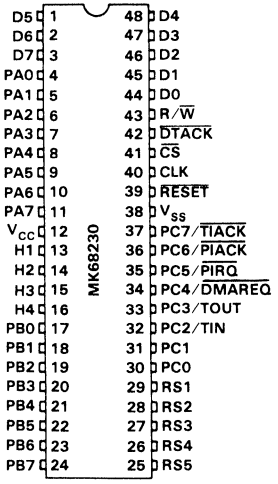
The PI/T consists of two logically independent sections: the ports and the timer. The port section consists of Port

MK68230
Figure 1



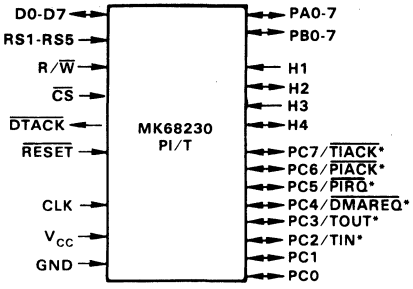
PIN ASSIGNMENT

Figure 2



LOGICAL PIN ASSIGNMENT

Figure 3



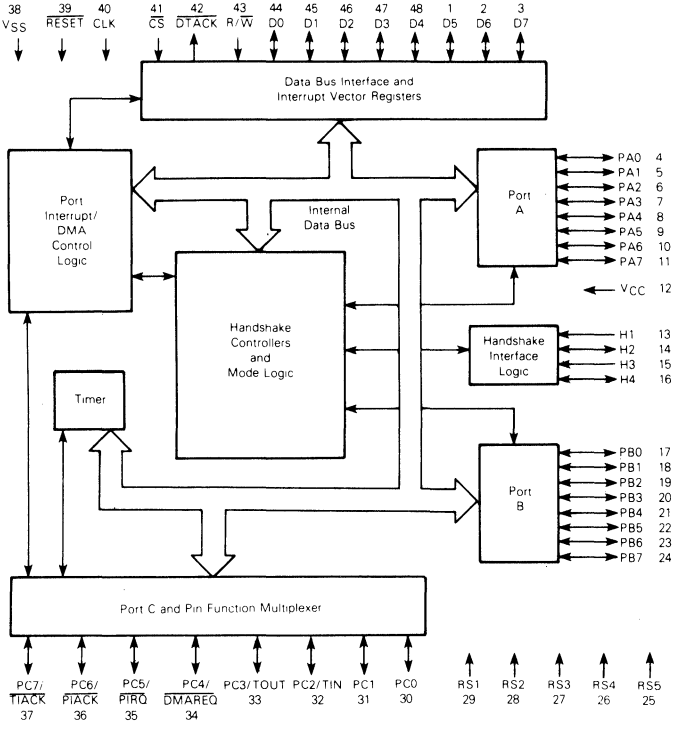
*Individually Programmable Dual-Function Pin

A (PA0-PA7), Port B (PB0-PB7), four handshake pins (H1, H2, H3, and H4), two general I/O pins, and six dual-function pins. The dual-function pins can individually operate as a third port (Port C) or as an alternate function related to either Ports A and B, or the timer. The four programmable handshake pins, depending on the mode, can control data transfer to and from the ports, can be used as interrupt generating inputs, or can be used as I/O pins. The timer consists of a 24-bit counter, optionally clocked by a 5-bit prescaler. Three pins provide complete timer I/O: PC2/TIN, PC3/TOUT, and PC7/TIACK. Of course, only the ones needed for the given configuration perform the timer function, while the others remain Port C I/O.

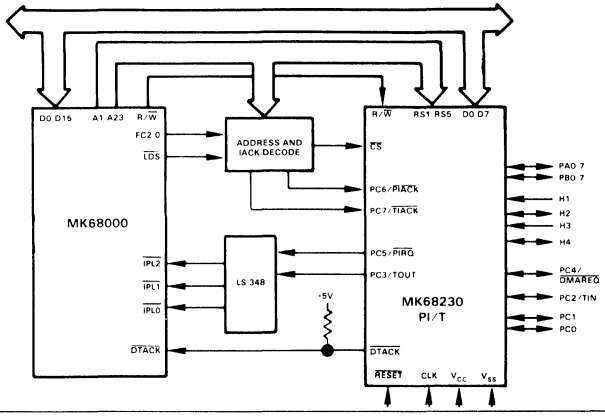
The system bus interface provides for asynchronous transfer of data from the PI/T to a bus master over the data bus (D0-D7). Data transfer acknowledge (DTACK), register selects (RS1-RS5), chip select, the read/write line (R/W), and Port Interrupt Acknowledge (PIACK) or Timer Interrupt Acknowledge (TIACK) control data transfer between the PI/T and the MK68000.

MK68320 BLOCK DIAGRAM

Figure 4



PI/T SYSTEM BLOCK DIAGRAM
Figure 5



PIN DESCRIPTION

Throughout this data sheet, signals are presented using the terms active and inactive, or asserted and negated independent of whether the signal is active in the high-voltage state or low-voltage state. (The active state of each logic pin is given below.) Active low signals are denoted by a superscript bar. R/W indicates a "write" is active low and a "read" active high.

D0-D7 (Bidirectional Data Bus) — The data bus pins D0-D7 form an 8-bit bidirectional data bus to/from the MK68000 or other bus master. These pins are active high.

RS1-RS5 (Register Selects) — RS1-RS5 are active high, high-impedance inputs that determine which of the 25 possible registers is being addressed. They are provided by the MK68000 or other bus master.

R/$\overline{W}$ (Read/$\overline{Write}$ Input) — R/$\overline{W}$ is the high-impedance Read/$\overline{Write}$ signal from the MK68000 or bus master, indicating whether the current bus cycle is a read (high cycle) or write (low cycle).

$\overline{CS}$ (Chip Select Input) — $\overline{CS}$ is a high-impedance input that selects the PI/T registers for the current bus cycle. Address strobe and the data strobe (upper and lower) of the bus master, along with the appropriate address bits, must be included in the chip select equation. A low level corresponds to an asserted chip select.

$\overline{DTACK}$ (Data Transfer Acknowledge Output) — $\overline{DTACK}$ is an active low output that signals the completion of the bus cycle. During read or interrupt acknowledge cycles, $\overline{DTACK}$ is asserted by the MK68230 after data has been provided on the data bus; during write cycles it is asserted after data has been accepted at the data bus. Data transfer acknowledge is compatible with the MK68000 and with other Mostek bus masters. A holding resistor is required to maintain $\overline{DTACK}$ high between bus cycles.

$\overline{RESET}$ (Reset Input) — $\overline{RESET}$ is a high-impedance input used to initialize all PI/T functions. All control and data direction registers are cleared and most internal operations are disabled by the assertion of $\overline{RESET}$ (low).

CLK (Clock Input) — The clock pin is a high-impedance, TTL-compatible signal with the same specifications as the MK68000. The PI/T contains dynamic logic throughout, and hence this clock must not be gated off at any time. It is not necessary that this clock maintain any particular phase relationship with the MK68000 clock. It may be connected to an independent frequency source (faster or slower) as long as all bus specifications are met.

PA0-PA7 and PB0-PB7 (Port A and Port B) — Ports A and B are 8-bit ports that may be concatenated to form a 16-bit port in certain modes. The ports may be controlled in conjunction with the handshake pins H1-H4. For stabilization during system power-up, Ports A and B have internal pull up resistors to Vcc. All port pins are active high.

H1-H4 (Handshake Pins Inputs or Output) — Handshake pins H1-H4 are multi-purpose pins that (depending on operational mode) may provide an interlocked handshake, a pulsed handshake, an interrupt input (independent of data transfers), or simple I/O pins. For stabilization during system power-

up, H-2 and H4 have internal pullup resistors to Vcc. Their sense (active high or low) may be programmed in the Port General Control Register bits 3-0. Independent of the mode, the instantaneous level of the handshake pins can be read from the Port Status Register.

Port C (PC0-PC7/ Alternate Function)

This port can be used as eight general purpose I/O pins (PC0-PC7) or any combination of six special function pins and two general purpose I/O pins (PC0-PC1). (Each dual function pin can be standard I/O or a special function independent of the other Port C pins.) The dual function pins are defined in the following paragraphs. When used as a Port C pin, these pins are active high. They may be individually programmed as inputs or outputs by the Port C Data Direction Register.

The alternate functions (TIN, TOUT, $\overline{\text{TIACK}}$) are timer I/O pins. TIN may be used as a rising-edge triggered external clock input or an external run/halt control pin (the timer is in the run state if run/halt is high and in the halt state if run/halt is low). TOUT may provide an active low timer interrupt request output or a general-purpose square-wave output, initially high. $\overline{\text{TIACK}}$ is an active low high-impedance input used for timer interrupt acknowledge.

Port A and B functions have an independent pair of active low interrupts request ($\overline{\text{PIRQ}}$) and interrupt acnkowledge ($\overline{\text{PIACK}}$) pins.

The $\overline{\text{DMAREQ}}$ (Direct Memory Access Request) pin provides an active low Direct Memory Access Controller (DMAC) request pulse of three clock cylcles.

MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature	T_{stg}	−55 to +150	°C

NOTE:
This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 5%, T_A = 0 to 70°C, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Input High Voltage — All inputs	V_{IH}	V_{SS} +2.0	V_{CC}	V
Input Low Voltage — All inputs	V_{IL}	V_{SS} −0.3	V_{SS} +0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V) H1, H3, R/$\overline{W}$, $\overline{RESET}$, CLK, RS1-RS5, $\overline{CS}$	I_{in}	—	10.0	μA
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4) $\overline{DTACK}$, PC0-PC7, D0-D7	I_{TSI}	—	20	μA
H2, H4, PA0-PA7, PB0-PB7		−0.1	−1.0	mA
Output High Voltage (I_{LOAD} = −400μA, V_{CC} = min) $\overline{DTACK}$, D0-D7 (I_{LOAD} = −150μA, V_{CC} = min) H2, H4, PB0-PB7, PA0-PA7 (I_{LOAD} = −100μA, V_{CC} = min) PC0-PC7	V_{OH}	V_{SS} +2.4	—	V
Output Low Voltage (I_{LOAD} = 8.8 mA, V_{CC} = min) PC3/TOUT, PC5/$\overline{PIRQ}$ (I_{LOAD} = 5.3 mA, V_{CC} = min) D0-D7, $\overline{DTACK}$ (I_{LOAD} = 2.4 mA, V_{CC} = min) PA0-PA7, PB0-PB7, H2, H4, PC0-PC2, PC4, PC6,PC7	V_{OL}	—	0.5	V
Internal Power Dissipation (Measured at T_A = 0°C)	P_{INT}	—	750	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1 MHz)	C_{in}	—	15	pF

AC ELECTRICAL SPECIFICATIONS — CLOCK TIMING

Characteristics	Symbol	8 MHz		10 MHz		12.5 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f	2.0	8.0	2.0	10.0	4.0	12.0	MHz
Cycle Time	t_{cyc}	125	500	100	500	80	250	ns
Clock Pulse Width	t_{CL}	55	250	45	250	35	125	ns
	t_{CH}	55	250	45	250	35	125	
Clock Rise and Fall Times	t_{Cr}	—	10	—	10	—	5	ns
	t_{Cf}	—	10	—	10	—	5	

AC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C, unless otherwise noted)

Number	Characteristics	8 MHz		10 MHz		12.5 MHz		Unit
		Min	Max	Min	Max	Min	Max	
1	R/$\overline{W}$, RS1-RS5 Valid to CS Low (Setup Time)	0	—	0	—	0	—	ns
2	$\overline{CS}$ Low to R/$\overline{W}$ and RS1-RS5 Invalid (Hold Time)	100	—	65	—	60	—	ns
3(1)	$\overline{CS}$ Low to CLK Low (Setup Time)	30	—	20	—	20	—	ns
4(2)	$\overline{CS}$ Low to Data Out Valid	—	75	—	65	—	55	ns
5	RS1-RS5, R/$\overline{W}$ Valid to Data Out	—	140	—	100	—	80	ns
6	CLK Low to $\overline{DTACK}$ Low (Read/Write Cycle)	0	70	0	60	0	55	ns
7(3)	$\overline{DTACK}$ Low to $\overline{CS}$ High (Hold Time)	0	—	0	—	0	—	ns
8	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to Data Out Invalid (Hold Time)	0	—	0	—	0	—	ns
9	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to D0-D7 High Impedance	—	50	—	45	—	45	ns

AC ELECTRICAL SPECIFICATIONS (Continued)

Number	Characteristic	8 MHz		10 MHz		12.5 MHz		Unit
		Min	Max	Min	Max	Min	Max	
10	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to $\overline{DTACK}$ High	—	50	—	45	—	40	ns
11	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to $\overline{DTACK}$ High Impedance	—	100	—	55	—	45	ns
12	Data In Valid to $\overline{CS}$ Low (Setup Time)	0	—	0	—	0	—	ns
13	$\overline{CS}$ Low to Data In Invalid (Hold Time)	100	—	65	—	60	—	ns
14	Port Input Data Valid to H1(H3) Asserted (Setup Time)	100	—	60	—	50	—	ns
15	H1(H3) Asserted to Port Input Data Invalid (Hold Time)	20	—	20	—	20	—	ns
16	Handshake Input H1(H4) Pulse Width Asserted	40	—	40	—	40	—	ns
17	Handshake Input H1(H4) Pulse Width Negated	40	—	40	—	40	—	ns
18	H1(H3) Asserted to H2(H4) Negated (Delay Time)	—	150	—	120	—	100	ns
19	CLK Low to H2(H4) Asserted (Delay Time)	—	100	—	100	—	80	ns
20(4)	H2(H4) Asserted to H1(H3) Asserted	0	—	0	—	0	—	ns
21(5)	CLK Low to H2(H4) Pulse Negated (Delay Time)	—	125	—	125	—	100	ns
22(9,10)	Synchronized H1(H3) to CLK Low on which $\overline{DMAREQ}$ is Asserted	2.5	3.5	2.5	3.5	2.5	3.5	CLK Per.
23	CLK Low on which $\overline{DMAREQ}$ is Asserted to CLK Low on which $\overline{DMAREQ}$ is Negated	2.5	3	2.5	3	2.5	3	CLK Per.
24	CLK Low to Port Output Data Valid (Delay Time) (Modes 0 and 1)	—	150	—	120	—	100	ns
25(9,10)	Synchronized H1(H3) to Port Output Data Invalid (Modes 0 and 1)	1.5	2.5	1.5	2.5	1.5	2.5	CLK Per.
26	H1 Negated to Port Output Data Valid (Modes 2 and 3)	—	70	—	50	—	50	ns
27	H1 Asserted to Port Output Data High Impedance (Modes 2 and 3)	0	70	0	70	0	70	ns
28	Read Data Valid to $\overline{DTACK}$ Low (Setup Time)	0	—	0	—	0	—	ns
29	CLK Low to Data Output Valid, Interrupt Acknowledge Cycle	—	120	—	100	—	80	ns
30(7)	H1(H3) Asserted to CLK High (Setup Time)	50	—	40	—	40	—	ns
31	$\overline{PIACK}$ or $\overline{TIACK}$ Low to CLK Low (Setup Time)	50	—	40	—	30	—	ns

AC ELECTRICAL SPECIFICATIONS (Continued)

Number	Characteristic	8 MHz		10 MHz		12.5 MHz		Unit
		Min	Max	Min	Max	Min	Max	
32(10)	Synchronized $\overline{CS}$ to CLK Low on which $\overline{DMAREQ}$ is Asserted	3	3	3	3	3	3	CLK Per.
33(9,10)	Synchronized H1(H3) to CLK Low on which H2(H4) is Asserted	3.5	4.5	3.5	4.5	3.5	4.5	CLK Per.
34	CLK Low to $\overline{DTACK}$ Low Interrupt Acknowledge Cycle (Delay Time)	—	100	—	100	—	80	ns
35	CLK Low to $\overline{DMAREQ}$ Low (Delay Time)	0	120	0	100	0	80	ns
36	CLK Low to $\overline{DMAREQ}$ High (Delay Time)	0	120	0	100	0	80	ns
37(10)	Synchronized H1(H3) to CLK Low on which $\overline{PIRQ}$ is Asserted	2.5	3.5	2.5	3.5	2.5	3.5	CLK Per.
38(10)	Synchronized $\overline{CS}$ to CLK Low on which $\overline{PIRQ}$ is High Impedance	3	3	3	3	3	3	CLK Per.
39	CLK Low to PIRQ Low or High Impedance	0	250	0	225	0	200	ns
40(8)	TIN Frequency (External Clock) - Prescaler Used	0	1	0	1	0	1	f_{clk} (Hz) (6)
41	TIN Frequency (External Clock) - Prescaler Not Used	0	1/8	0	1/8	0	1/8	f_{clk} (Hz) (6)
42	TIN Pulse Width High or Low (External Clock)	55	—	45	—	45	—	ns
43	TIN Pulse Width Low (Run/Halt Clock)	1	—	1	—	1	—	CLK Per.
44	CLK Low to TOUT High, Low, or High Impedance	0	250	0	225	0	200	ns
45	$\overline{CS}$, $\overline{PIACK}$, or $\overline{TIACH}$ High to $\overline{CS}$, $\overline{PIACK}$, or $\overline{TIACK}$ Low	50	—	30	—	30	—	ns

NOTES:

1. This specification only applies if the PI/T had completed all operations initiated by the previous bus cycle when $\overline{CS}$ was asserted. Following a normal read or write bus cycle, all operations are complete within three clocks after the falling edge of the CLK pin on which $\overline{DTACK}$ was asserted. If $\overline{CS}$ is asserted prior to completion of these operations, the new bus cycle, and hence, $\overline{DTACK}$ is postponed.

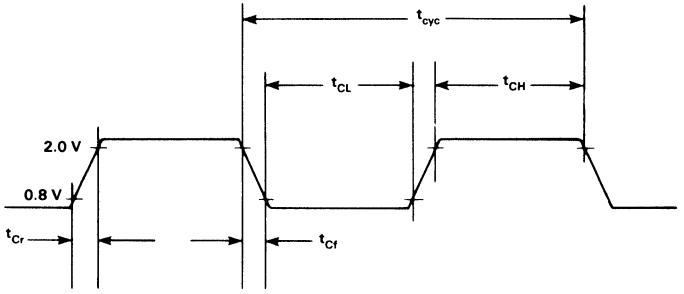
   If all operations of the previous bus cycle were complete when $\overline{CS}$ was asserted, this specification is made only to insure that $\overline{DTACK}$ is asserted with respect to the falling edge of the CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the $\overline{CS}$ setup time is violated, $\overline{DTACK}$ may be asserted as shown, or may be asserted one clock cycle later.
2. Assuming the RS1-RS5 to data valid time has also expired.
3. This specification imposes a lower bound on $\overline{CS}$ low time, guaranteeing that $\overline{CS}$ will be low for at least 1 CLK period.
4. This specification assures recognition of the asserted edge of H1(H3).
5. This specification applies only when a pulsed handshake option is chosen and the pulse is not shortened due to an early asserted edge of H1(H3).
6. CLK refers to the actual frequency of the CLK pin, not the maximum allowable CLK frequency.
7. If the setup time on the rising edge of the clock is not met, H1(H3) may not be recognized until the next rising of the clock.
8. This limit applies to the frequency of the signal at TIN compared to the frequency of the CLK signal during each clock cycle. If any period of the waveform at TIN is smaller than the period of the CLK signal at that instant, then it is likely that the timer circuit will completely ignore one cycle of the TIN signal.

   If these two signals are derived from different sources, they will have different instantaneous frequency variations. In this case the frequency applied to the TIN pin must be distinctly less than the frequency at the CLK pin to avoid lost cycles of the TIN signal. With signals derived from different crystal oscillators applied to the TIN and CLK pins with fast rise and fall times, the TIN frequency can approach 80 to 90% of the frequency of the CLK signal without a loss of a cycle of the TIN signal.

   If these two signals are derived from the same frequency source then the frequency of the signal applied to TIN can be 100% of the frequency at the CLK pin. They may be generated by different buffers from the same signal or one may be an inverted version of the other. The TIN signal may be generated by and 'AND' function of the clock and a control signal.
9. The maximum value is caused by a peripheral access (H1 (H3) asserted) and bus access ($\overline{CS}$ asserted) occurring at the same time.
10. Synchronized means that the input signal has been seen by the PI/T on the appropriate edge of the clock (rising edge for H1(H3) and falling edge for $\overline{CS}$).

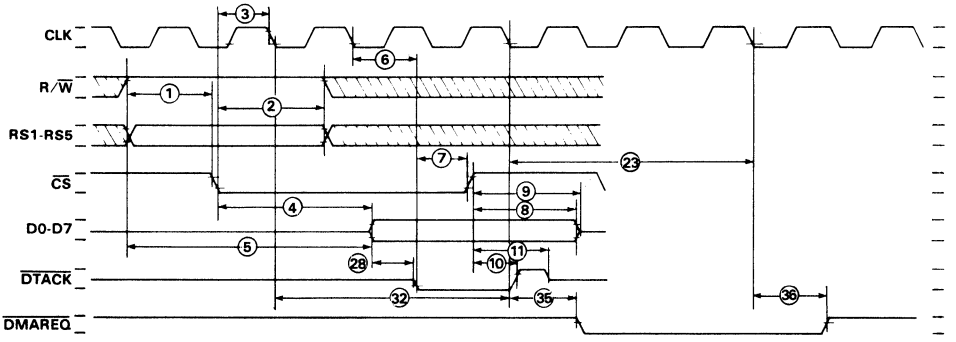
CLOCK INPUT TIMING DIAGRAM

Figure 6



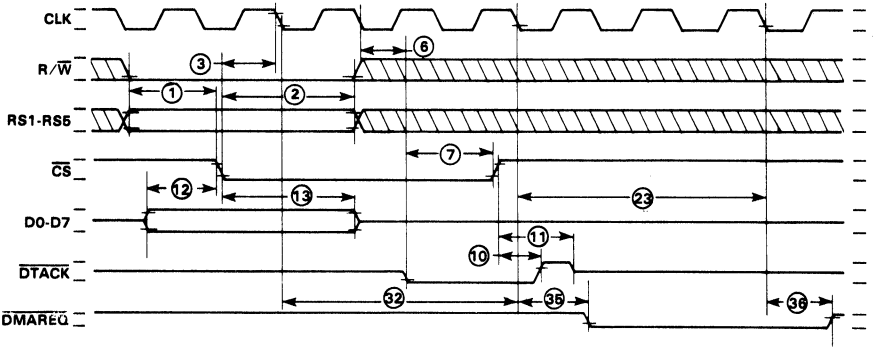
READ CYCLE TIMING DIAGRAM

Figure 7



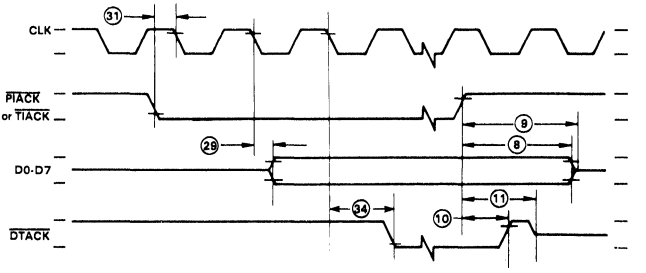
WRITE CYCLE TIMING DIAGRAM

Figure 8



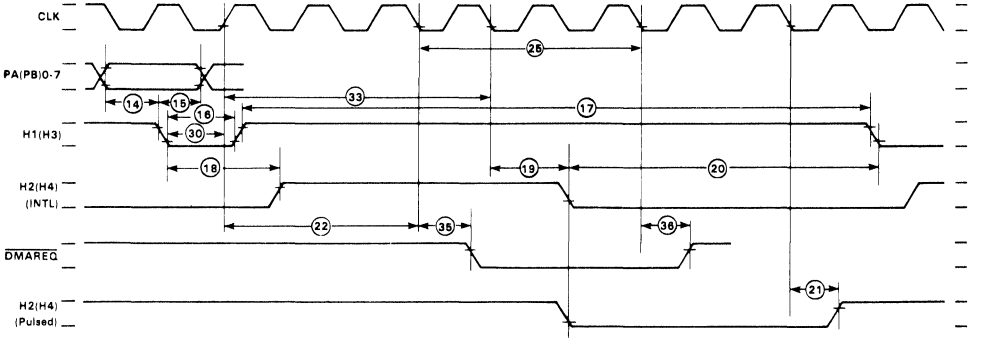
NOTE:
Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

IACK TIMING DIAGRAM
Figure 9

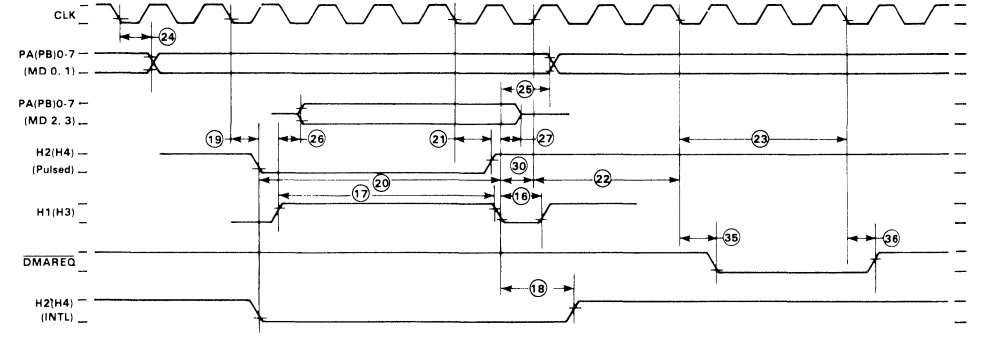


NOTE:
Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise needed.

PERIPHERAL INPUT TIMING DIAGRAM
Figure 10



PERIPHERAL OUTPUT TIMING DIAGRAM
Figure 11



NOTES:
1. Timing diagram shows H1, H2, H3, and H4 asserted low.
2. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

MK68230 ORDERING INFORMATION

PART NO.	PACKAGE TYPE	MAX CLOCK FREQUENCY	TEMP. RANGE
MK68230N-8	48 Pin Plastic DIP	8.0 MHz	0° to 70°C
MK68230N-10		10.0 MHz	

THOMSON COMPONENTS MOSTEK

MK68451

MEMORY MANAGEMENT UNIT

ADVANCE INFORMATION

MICROCOMPUTER COMPONENTS

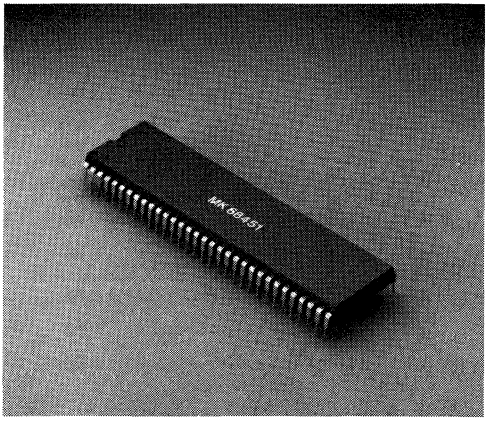
FEATURES

- ☐ Compatible with MK68000 and MK68008
- ☐ Provides virtual memory support for the MK68010
- ☐ Provides efficient memory allocation
- ☐ Seperates address spaces of system and user resources
- ☐ Provides write protection
- ☐ Supports paging and segmentation
- ☐ 32 segments of variable size with each MMU
- ☐ Multiple MMU capability to expand to any number of segments
- ☐ Allows inter-task communication through shared segments
- ☐ Quick context switching to cut operating system overhead
- ☐ Simplifies programming model of address space
- ☐ Increases system reliability
- ☐ DMA-compatible

GENERAL DESCRIPTION

The MK68451 memory management unit (MMU) provides address translation and protection for the 16 megabyte addressing range of the MK68000 MPU. Each bus master (or processor) in the MK68000 family provides a function code and an address during each bus cycle. The function code specifies an address space, and the address specifies a location within that address space. The function codes distinguish between user and supervisor spaces and, within these, between data and program spaces. This separation of address spaces provides the basis for memory management and protection by the operating system. Provision is also made for other bus masters to have separate address spaces for logical DMA.

MK68451
Figure 1



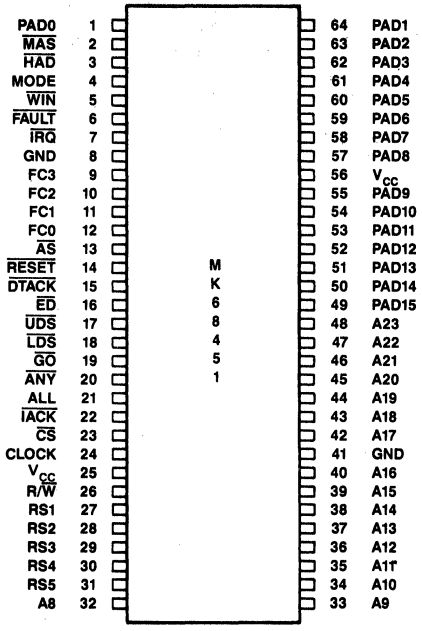
A multitasking operating system is simplified, and reliability is enhanced, through the use of the MMU.

The MK68451 memory management unit (MMU) is the basic element of a memory management mechanism (MMM) in an MK68000 family system. The operating system is responsible for insuring the proper execution of user tasks in the system environment, and memory management is basic to this responsibility. The MMM provides the operating system with the capability to allocate, control, and protect the system memory. A block diagram of a single-MMU system is shown in Figure 3.

An MMM, implemented with one or more MK68451 MMUs, can provide address translation, separation, and write protection for the system memory. The MMM can be programmed to cause an interrupt when a chosen section of memory is accessed, and can directly translate a logical address into a physical address, making it available to the MPU for use by the operating system. Using these features, the MMM can provide separation and security for user programs and allow the operating system to manage the memory in an efficient fashion for multitasking.

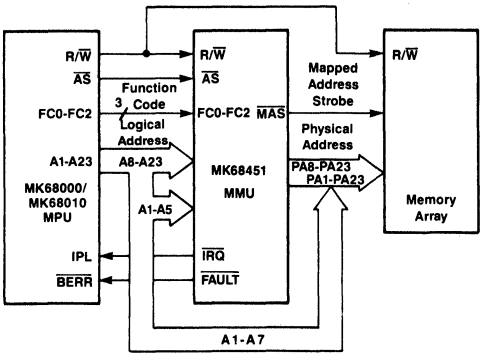
PIN ASSIGNMENT

Figure 2



SIMPLIFIED BLOCK DIAGRAM OF SINGLE-MMU SYSTEM

Figure 3



FUNCTIONAL DESCRIPTION

MEMORY SEGMENTS

The MMM partitions the logical address space into contiguous pieces called segments. Each segment is a section of the logical address space of a task which is mapped via the MMM into the physical address space. Each task may have any number of segments. Segments may be defined as user or supervisor, data-only or program-only, or program and data. They may be accessed by only one task or shared between two or more tasks. In addition, any segment can be write protected to insure system integrity. A fault (MK68000 bus error) is generated by the MMM if an undefined segment is accessed.

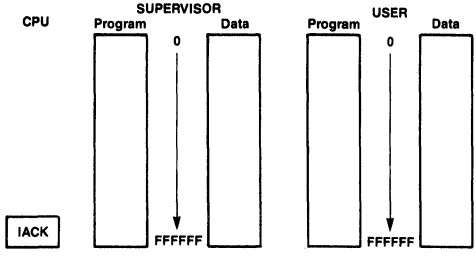
FUNCTION CODES AND ADDRESS SPACES

Each bus master in the MK68000 family provides a function code during each bus cycle to indicate the address space to be used for that cycle. The address bus then specifies a location within this address space for the operation taking place during that bus cycle.

The function codes appear on the FC0-FC2 lines of the MK68000 and divide the memory references into two logical address spaces—the supervisor and the user spaces. Each of these is further divided into program and data spaces. A separate address space is also provided for internal CPU-related activities, such as interrupt acknowledge, giving a total of five defined function codes. The address space of the MK68000 is shown in Figure 4.

ADDRESS SPACE OF MK68000

Figure 4



In addition to the 3-bit function code provided by the MK68000, the MK68451 MMU also allows a fourth bit (FC3) which provides for the possibility of another bus master in the system. In this case, FC3 would be a function of bus grant acknowledge (BGACK) of the MK68000 to enable a second set of eight function codes. This raises the total number of possible function codes to 16. If there

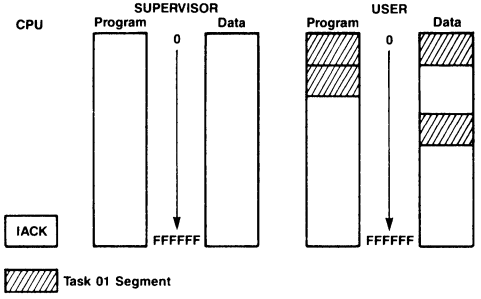
is only one bus master (the MPU), the FC3 pin on the MMU should be tied low, and only eight address spaces can then be used.

ADDRESS SPACE NUMBER

Each task in a system has an address space comprised of all the segments defined for that task. This address space is assigned a number by programming all the address space number (ASN) fields in its descriptors with the same value. This value can be considered a task number. The currently active task's number is kept in the appropriate entry(s) in the address space table (AST).

The AST is a set of MMU registers that defines which task's segments are to be used in address translation for each cycle type (supervisor program, supervisor data, etc.). The AST contains an 8-bit entry for each possible function code. Each entry is assigned an ASN (task number) and this is used to select which descriptors may be used for translation. The logical address is then translated by one of these to produce the physical address. Figure 5 is a typical memory map of a task's address space.

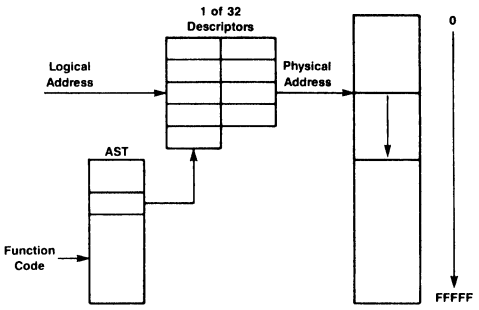
MEMORY MAP OF TYPICAL TASK ADDRESS SPACE
Figure 5



DESCRIPTORS

Address translation is done using descriptors. A descriptor is a set of six registers (nine bytes) which describe a memory segment and how that segment is to be mapped to the physical addresses. Each descriptor contains base addresses for the logical address masks. The size of the segment is then defined by "don't cares" in the masks. This method allows segment sizes from a minimum of 256 bytes to a maximum of 16 megabytes in binary increments (i.e., powers of two). This also forces both logical and physical addresses of segment boundaries to lie on a segment size boundary. That is, a segment can only start on an address which is a multiple of 2k. The segments can be defined in such a way to allow them to be logically or physically shared between tasks. Descriptor mapping is shown schematically in Figure 6.

SCHEMATIC DIAGRAM OF DESCRIPTOR MAPPING
Figure 6

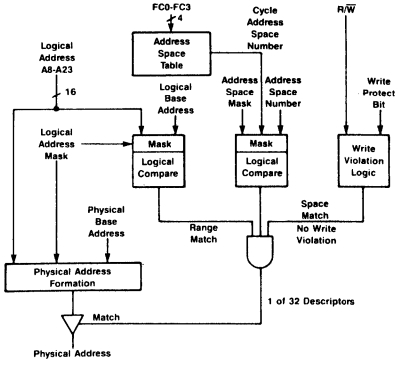


TRANSLATION

During normal translation, the MMU translates the logical address provided by the MK68000 to produce a physical address which is then presented to the memory array. This is accomplished by matching the logical address with the information in the descriptors and then mapping it into the physical address space. A block diagram of the MK68451 is shown in Figure. 7

Refer to Figure 3 for the following information. The logical address is composed of address lines A1-A23. The upper 16 bits of this address (A8-A23) are translated by the MMU and mapped into a physical address (PA8-PA23). The lower seven bits of the logical address (A1-A7) bypass the MMU and become the low-order physical address bits (PA1-PA7). In addition, the data strobes ($\overline{UDS}$ and $\overline{LDS}$) remain unmapped to become the physical data strobes for a total of eight unmapped address lines.

FUNCTIONAL BLOCK DIAGRAM
Figure 7

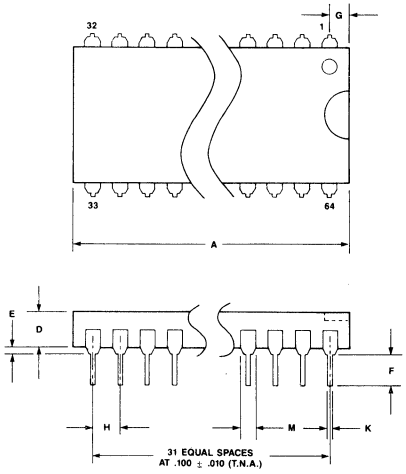


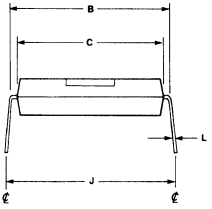
ORDERING INFORMATION

Part Number	Package Type	Max Clock Frequency	Temperature Range
MK68451N-8	Plastic	8.0 MHz	0° to 70°C
MK68451N-10	Plastic	10.0 MHz	0° to 70°C

PACKAGE DESCRIPTION

Dual In-Line
64-Pin





DIM.	INCHES		NOTES
	MIN.	MAX.	
A	3.180	3.230	2
B	.890	.940	
C	.790	.810	
D	.170	.190	
E	.020	.060	3
F	.120	.150	
G	.040	.070	
H	.090	.110	
J	.900	1.000	4
K	.015	.021	5
L	.008	.012	5
M	.045	.070	

NOTES;

1. LEAD FINISH IS TO BE SPECIFIED ON THE DETAIL SPECIFICATION.
2. OVERALL LENGTH INCLUDES .010 IN. FLASH ON EITHER END OF THE PACKAGE.
3. PACKAGE STANDOFF TO BE MEASURED PER JEDEC REQUIREMENTS.
4. MEASURED FROM CENTERLINE TO CENTERLINE AT LEAD TIPS.
5. WHEN THE SOLDER LEAD FINISH IS SPECIFIED, THE MAXIMUM LIMIT SHALL BE INCREASED BY .003 IN.

THOMSON COMPONENTS MOSTEK

MK68564

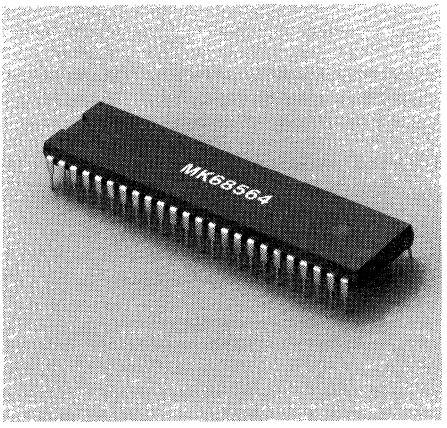
SERIAL INPUT/OUTPUT CONTROLLER

PRELIMINARY — MICROCOMPUTER COMPONENTS

FEATURES

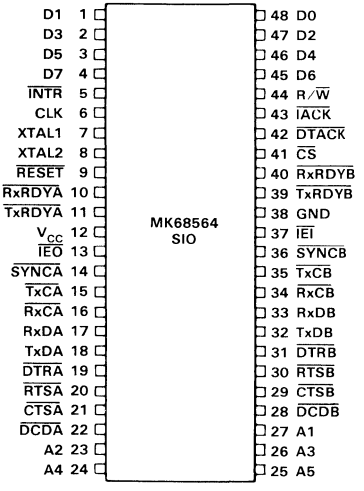
- ☐ Compatible with MK68000 CPU
- ☐ Compatible with MK68000 Series DMA's
- ☐ Two independent, full-duplex channels
- ☐ Two independent baud rate generators
  - Crystal oscillator input
  - Single-phase TTL clock input
- ☐ Directly addressable registers (all control registers are read/write)
- ☐ Data rate in synchronous or asynchronous modes
  - 0-1.25 M bits/second with 5.0 MHz system clock rate
- ☐ Self-test capability
- ☐ Receive data registers are quadruply buffered; transmit data registers are doubly buffered
- ☐ Daisy-chain priority interrupt logic provides automatic interrupt vectoring without external logic
- ☐ Modem status can be monitored
  - Separate modem controls for each channel
- ☐ Asynchronous features
  - 5, 6, 7, or 8 bits/character
  - 1, 1½, or 2 stop bits
  - Even, odd, or no parity
  - x1, x16, x32, and x64 clock modes
  - Break generation and detection
  - Parity, overrun, and framing error detection
- ☐ Byte synchronous features
  - Internal or external character synchronization
  - One or two sync characters in separate registers
  - Automatic sync character insertion
  - CRC-16 or CRC-CCITT block check generation and checking
- ☐ Bit synchronous features
  - Abort sequence generation and detection
  - Automatic zero insertion and deletion
  - Automatic flag insertion between messages
  - Address field recognition
  - I-field residue handling
  - Valid receive messages protected from overrun
  - CRC-CCITT block check generation and checking

MK68564
Figure 1



PIN DESCRIPTION

Figure 2



GENERAL DESCRIPTION

The MK68564 SIO is a dual-channel, Serial Input/Output Controller, designed to satisfy a wide variety of serial data communications requirements in microcomputer systems. Its basic function is a serial-to-parallel, parallel-to-serial converter/controller; however, within that role, it is systems software configurable so that it may be optimized for any given serial data communications application.

The MK68564 is capable of handling asynchronous protocols, synchronous byte-oriented protocols (such as IBM Bisync), and synchronous bit-oriented protocols (such as HDLC and IBM SDLC). This versatile device can also be used to support virtually any serial protocol for applications other than data communications (cassette or floppy disk interface, for example).

The MK68564 can generate and check CRC codes in any synchronous mode and may be programmed to check data integrity in various modes. The device also has facilities for modem controls in each channel. In applications where these controls are not needed, the modem controls may be used for general-purpose I/O.

SIO PIN DESCRIPTION

GND:	Ground.
V_{CC}:	+5 volts (±5%).
$\overline{CS}$: Chip Select	Input active low. $\overline{CS}$ is used to select the MK68564 SIO for access to the internal registers. $\overline{CS}$ and $\overline{IACK}$ must not be asserted at the same time.
R/$\overline{W}$: Read/Write	Input. R/$\overline{W}$ is the signal from the bus master, indicating whether the current bus cycle is a read (high) or write (low) cycle.
$\overline{DTACK}$: Data Transfer Acknowledge	Output, active low, tri-stateable. $\overline{DTACK}$ is used to signal the bus master that data is ready or that data has been accepted by the MK68564 SIO.
A1-A5: Address Bus	Inputs. The address bus is used to select one of the internal registers during a read or write cycle.
D0-D7: Data Bus	Bidirectional, tri-stateable. The data bus is used to transfer data to or from the internal registers during a read or write cycle. It is also used to pass a vector during an interrupt acknowledge cycle.
CLK: Clock	Input. This input is used to provide the internal timing for the MK68564 SIO.
$\overline{RESET}$ Device Reset	Input, active low. $\overline{Reset}$ disables both receivers and transmitters, forces TxDA and TxDB to a marking condition, forces the modem controls high, and disables all interrupts. With the exception of the status registers, data registers, and the vector register, all internal registers are cleared. The vector register is reset to "0FH".
$\overline{INTR}$ Interrupt Request	Output, active low, open drain. $\overline{INTR}$ is asserted when the MK68564 SIO is requesting an interrupt. $\overline{INTR}$ is negated during an interrupt acknowledge cycle or by clearing the pending interrupt(s) through software.
$\overline{IACK}$ Interrupt Acknowledge	Input, active low. $\overline{IACK}$ is used to signal the MK68564 SIO that the CPU is acknowledging an interrupt. $\overline{CS}$ and $\overline{IACK}$ must not be asserted at the same time. If interrupts are not used then $\overline{IACK}$ should be pulled high.
$\overline{IEI}$ Interrupt Enable In	Input, active low. $\overline{IEI}$ is used to signal the MK68564 SIO that no higher priority device is requesting interrupt service.
$\overline{IEO}$ Interrupt Enable Out	Output, active low. $\overline{IEO}$ is used to signal lower priority peripherals that neither the MK68564 SIO nor another higher priority peripheral is requesting interrupt service.
XTAL1 XTAL2 Baud Rate Generator Inputs	Inputs.A crystal may be connected between XTAL1 and XTAL2, or XTAL1 may be driven with a TTL level clock. When using a crystal, external capacitors must be connected. When driving XTAL1 with a TTL level clock, XTAL2 must be allowed to float.
$\overline{RxRDYA}$ $\overline{RxRDYB}$ Receiver Ready	Outputs, active low. Programmable DMA output for the receiver. The $\overline{RxRDY}$ pins pulse low when a character is available in the receive buffer.
$\overline{TxRDYA}$ $\overline{TxRDYB}$ Transmitter Ready	Outputs, active low. Programmable DMA output for the transmitter. The $\overline{TxRDY}$ pins pulse low when the transmit buffer is empty.
$\overline{CTSA}$ $\overline{CTSB}$ Clear to Send	Inputs, active low. If Tx Auto Enables is selected, these inputs enable the transmitter of their respective channels. If Tx Auto Enables is not selected, these inputs may be used as general purpose input pins. The inputs are Schmitt-trigger buffered to allow slow rise-time input signals.

$\overline{\text{DCDA}}$ $\overline{\text{DCDB}}$ Data Carrier Detect	Inputs, active low. If Rx Auto Enables is selected, these inputs enable the receiver of their respective channels. If Rx Auto Enable is not selected, these inputs may be used as general purpose input pins. The inputs are Schmitt-trigger buffered to allow slow rise-time input signals.
RxDA RxDB Receive Data	Inputs, active high. Serial data input to the receiver.
TxDA TxDB Transmit Data	Outputs, active high. Serial data output of the transmitter.
$\overline{\text{RxCA}}$ $\overline{\text{RxCB}}$ Receiver Clocks	Input/output. Programmable pin, receive clock input, or baud rate generator output. The inputs are Schmitt-trigger buffered to allow slow rise-time input signals.
$\overline{\text{TxCA}}$ $\overline{\text{TxCB}}$ Transmitter Clocks	Input/output. Programmable pin, transmit clock input, or baud rate generator output. The inputs are Schmitt-trigger buffered to allow slow rise-time input signals.
$\overline{\text{RTSA}}$ $\overline{\text{RTSB}}$ Request to Send	Outputs, active low. These outputs follow the inverted state programmed into the RTS bit. When the RTS bit is reset in the asynchronous mode, the output will not change until the character in the transmitter is completely shifted out. These pins may be used as general purpose outputs.
$\overline{\text{DTRA}}$ $\overline{\text{DTRB}}$ Data Terminal Ready	Outputs, active low. These outputs follow the inverted state programmed into the DTR bit. These pins may also be used as general purpose outputs.
$\overline{\text{SYNCA}}$ $\overline{\text{SYNCB}}$ Synchronization	Input/output, active low. The $\overline{\text{SYNC}}$ pin is an output when Monosync, Bisync, or SDLC mode is programmed. It is asserted when a sync/flag character is detected by the receiver. The $\overline{\text{SYNC}}$ pin is a general purpose input in the Asynchronous mode and an input to the receiver in the External Sync mode.

MK68564 ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −25°C to +100°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with Respect to Ground ..... −.3 V to +7 V
Power Dissipation ..... 1.5 Watt

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 V ± 5%, GND = 0 Vdc, T_A = 0 to 70°C)

CHARACTERISTIC	SYM	MIN	MAX	UNIT
INPUT HIGH VOLTAGE ALL INPUTS	V_{IH}	GND + 2.0	V_{CC}	V
INPUT LOW VOLTAGE ALL INPUTS	V_{IL}	GND −0.3	GND +0.8	V
POWER SUPPLY CURRENT OUTPUTS OPEN	I_{LL}		190	mA
INPUT LEAKAGE CURRENT (V_{IN} = 0 to 5.25)	I_{IN}		±10	μA
THREE-STATE (OFF STATE) INPUT CURRENT $0 < V_{IN} < V_{CC}$ $\overline{DTACK}$, D0-D7, $\overline{SYNC}$, $\overline{TxC}$, $\overline{RxC}$, $\overline{INTR}$	I_{TSI}		20 ±10	μA μA
OUTPUT HIGH VOLTAGE (I_{LOAD} = −400 μA, V_{CC} = MIN) $\overline{DTACK}$, D0-D7 (I_{LOAD} = −150μA, V_{CC} = MIN) ALL OTHER OUTPUTS (EXCEPT XTAL2 & $\overline{INTR}$)*	V_{OH}	GND+2.4		V
OUTPUT LOW VOLTAGE (I_{LOAD} = 5.3 mA, V_{CC} = MIN) $\overline{INTR}$, $\overline{DTACK}$, D0-D7 (I_{LOAD} = 2.4 mA, V_{CC} = MIN) ALL OTHER OUTPUTS (EXCEPT XTAL2)*	V_{OL}		0.5	V

*XTAL2 SPECIAL
$\overline{INTR}$ (OPEN DRAIN)

CAPACITANCE

T_A = 25°C, f = 1 MHz unmeasured pins returned to ground.

CHARACTERISTIC	SYM	MAX	UNIT	TEST CONDITION
Input Capacitance $\overline{CS}$, $\overline{IACK}$ ALL OTHERS	C_{IN}	15 10	pf pf	Unmeasured pins returned to ground
Tri-state Output Capacitance	C_{OUT}	10	pf	

AC ELECTRICAL CHARACTERISTICS
(V_{CC}=5.0 Vdc±5%, GND=0 Vdc, T_A=0 to 70°C)

		3.0 MHz		4.0 MHz		5.0 MHz			
NUMBER	PARAMETER	MIN	MAX	MIN	MAX	MIN	MAX	UNITS	NOTES
1	CLK Period	330	1000	250	1000	200	1000	ns	
2	CLK Width High	145		105		80		ns	
3	CLK Width Low	145		105		80		ns	
4	CLK Fall Time		30		30		30	ns	
5	CLK Rise Time		30		30		30	ns	
6	$\overline{CS}$ Low to CLK High (Setup Time)	0		0		0		ns	1
7	A1-A5 Valid to $\overline{CS}$ Low (Setup Time)	0		0		0		ns	
8	DATA Valid to $\overline{CS}$ Low (Write Cycle)	0		0		0		ns	
9	$\overline{CS}$ Width High	50		50		50		ns	1
10	$\overline{DTACK}$ Low to A1-A5 Invalid (Hold Time)	0		0		0		ns	
11	$\overline{DTACK}$ Low to DATA Invalid (Write Cycle Hold Time)	0		0		0		ns	
12	$\overline{CS}$ High to $\overline{DTACK}$ High (Delay)		60		55		50	ns	
13	CLK High to $\overline{DTACK}$ Low		325		320		295	ns	
14	R/$\overline{W}$ Valid to $\overline{CS}$ Low (Setup Time)	0		0		0		ns	
15	$\overline{DTACK}$ Low to R/$\overline{W}$ Invalid (Hold Time)	0		0		0		ns	
16	CLK Low to DATA Out		550		450		450	ns	
17	$\overline{CS}$ High to DATA Out Invalid (Hold Time)	0		0		0		ns	11
18	$\overline{CS}$ High to $\overline{DTACK}$ High Impedance		110		105		100	ns	
19	$\overline{DTACK}$ Low to $\overline{CS}$ High	0		0		0		ns	
20	DATA Valid to $\overline{DTACK}$ Low	70		70		70		ns	
21	$\overline{IACK}$ Width High	50		50		50		ns	1
22	$\overline{IACK}$ Low to CLK High (Setup Time)	0		0		0		ns	1
23	CLK Low to $\overline{INTR}$ Disabled		410		410		410	ns	2
24	CLK Low to DATA Out		330		330		330	ns	2
25	$\overline{DTACK}$ Low to $\overline{IACK}$, $\overline{IEI}$ High	0		0		0		ns	
26	$\overline{IACK}$ High to $\overline{DTACK}$ High		60		55		50	ns	
27	$\overline{IACK}$ High to $\overline{DTACK}$ High Impedance		110		105		100	ns	
28	$\overline{IACK}$ High to DATA Out Invalid (Hold Time)	0		0		0		ns	
29	DATA Valid to $\overline{DTACK}$ Low	195		195		195		ns	2
30	CLK Low to $\overline{IEO}$ Low		220		220		220	ns	3

AC ELECTRICAL CHARACTERISTICS (Cont.)
(V_{CC}=5.0 Vdc±5%, GND=0 Vdc, T_A=0 to 70°C)

		3.0 MHz		4.0 MHz		5.0 MHz			
NUMBER	PARAMETER	MIN	MAX	MIN	MAX	MIN	MAX	UNITS	NOTES
31	$\overline{IEI}$ Low to $\overline{IEO}$ Low		140		140		140	ns	3
32	$\overline{IEI}$ High to $\overline{IEO}$ High		190		190		190	ns	4
33	$\overline{IACK}$ High to $\overline{IEO}$ High		190		190		190	ns	4
34	$\overline{IACK}$ High to $\overline{INTR}$ Low		200		200		200	ns	5
35	$\overline{IEI}$ Low to CLK Low (Setup Time)	10		10		10		ns	
36	$\overline{IEI}$ Low to $\overline{INTR}$ Disabled		500		425		425	ns	6
37	$\overline{IEI}$ Low to DATA Out Valid		225		225		225	ns	6
38	DATA Out Valid to $\overline{DTACK}$ Low	55		55		55		ns	6
39	$\overline{IACK}$ High to DATA Out High Impedance		150		120		90	ns	
40	$\overline{CS}$ High to DATA Out High Impedance		150		120		90	ns	
41	$\overline{CS}$ or $\overline{IACK}$ High to CLK Low	100		100		100		ns	7
42	$\overline{TxRDY}$ or $\overline{RxRDY}$ Width Low		3		3		3	CLK Period	8,10
43	CLK High to $\overline{TxRDY}$ or $\overline{RxRDY}$ Low		300		300		300	ns	
44	CLK High to $\overline{TxRDY}$ or $\overline{RxRDY}$ High		335		300		300	ns	
	$\overline{IACK}$ High to $\overline{CS}$ Low or $\overline{CS}$ High to $\overline{IACK}$ Low (not shown)	50		50		50		ns	1
45	$\overline{CTS}$, $\overline{DCD}$, $\overline{SYNC}$ Pulse Width High	200		200		200		ns	
46	$\overline{CTS}$, $\overline{DCD}$, $\overline{SYNC}$ Pulse Width Low	200		200		200		ns	
47	$\overline{TxC}$ Period	1320	DC	1000	DC	800	DC	ns	9
48	$\overline{TxC}$ Width Low	180	DC	180	DC	180	DC	ns	
49	$\overline{TxC}$ Width High	180	DC	180	DC	180	DC	ns	
50	$\overline{TxC}$ Low to TxD Delay (X1 Mode)		300		300		300	ns	
51	$\overline{TxC}$ Low to $\overline{INTR}$ Low Delay	5	9	5	9	5	9	CLK Period	10
52	$\overline{RxC}$ Period	1320	DC	1000	DC	800	DC	ns	9
53	$\overline{RxC}$ Width Low	180	DC	180	DC	180	DC	ns	
54	$\overline{RxC}$ Width High	180	DC	180	DC	180	DC	ns	
55	RxD to $\overline{RxC}$ High Setup Time (X1 Mode)	0		0		0		ns	
56	$\overline{RxC}$ High to RxD Hold Time (X1 Mode)	140		140		140		ns	
57	$\overline{RxC}$ High to $\overline{INTR}$ Low Delay	10	13	10	13	10	13	CLK Period	10
58	$\overline{RxC}$ High to $\overline{SYNC}$ Low Delay (Output Modes)	4	7	4	7	4	7	CLK Period	10

AC ELECTRICAL CHARACTERISTICS (Cont.)

(V_{CC}=5.0 Vdc±5%, GND=0 Vdc, T_A=0 to 70°C)

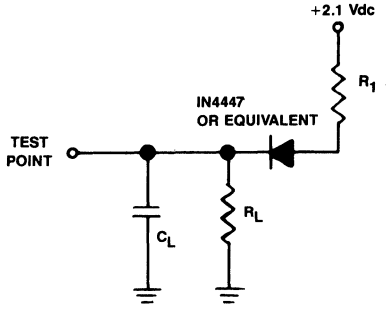
NUMBER	PARAMETER	3.0 MHz		4.0 MHz		5.0 MHz		UNITS	NOTES
		MIN	MAX	MIN	MAX	MIN	MAX		
59	$\overline{\text{RESET}}$ Low	1		1		1		CLK Period	10
60	XTAL 1 Width High (TTL in)	145		100		80		ns	
61	XTAL 1 Width Low (TTL in)	145		100		80		ns	
62	XTAL 1 Period (TTL in)	330	2000	250	2000	200	2000	ns	
63	XTAL 1 Period (Crystal in)	330	1000	250	1000	200	1000	ns	

NOTES:

1. This specification only applies if the SIO has completed all operations initiated by the previous bus cycle, when $\overline{CS}$ or $\overline{IACK}$ was asserted. Following a read, write, or interrupt acknowledge cycle, all operations are complete within two CLK cycles after the rising edge of $\overline{CS}$ or $\overline{IACK}$. If $\overline{CS}$ or $\overline{IACK}$ is asserted prior to the completion of the internal operations, the new bus cycle will be postponed.
2. If IEI meets the setup time to the falling edge of CLK, 1½ cycles following the clocking in of $\overline{IACK}$.
3. No internal interrupt request pending at the start of an interrupt acknowledge cycle.
4. Time starts when first signal goes invalid (high).
5. If an internal interrupt is pending at the end of the interrupt acknowledge cycle.
6. If Note 2 timing is not met.
7. If this spec is met, the delay listed in note 1 will be one CLK cycle instead of two.
8. Ready signals will be negated asynchronous to the CLK, if the condition causing the assertion of the signals is cleared.
9. If $\overline{RxC}$ and $\overline{TxC}$ are asynchronous to the System Clock, the maximum clock rate into $\overline{RxC}$ and $\overline{TxC}$ should be no more than one-fifth the System Clock rate. If $\overline{RxC}$ and $\overline{TxC}$ are synchronized to the falling edge of the System Clock, the maximum clock rate into $\overline{RxC}$ and $\overline{TxC}$ can be one-fourth the System Clock rate.
10. SIO Clock (CLK) Cycles as defined in Parameter 1.
11. Due to the dynamic nature of the internal data bus, if $\overline{CS}$ is held low for more than a few hundred milliseconds the read data may go to 00H before the end of the cycle.

OUTPUT TEST LOAD

Figure 3



for all outputs except $\overline{\text{DTACK}}$, D0-D7, $\overline{\text{INTR}}$, XTAL2

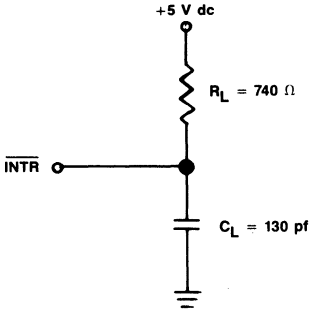
C_L = 130 pf
R_L = 16K Ω
R_1 = 450 Ω

for $\overline{\text{DTACK}}$, D0-D7

C_L = 130 pf
R_L = 6K Ω
R_1 = 200 Ω

$\overline{\text{INTR}}$ TEST LOAD

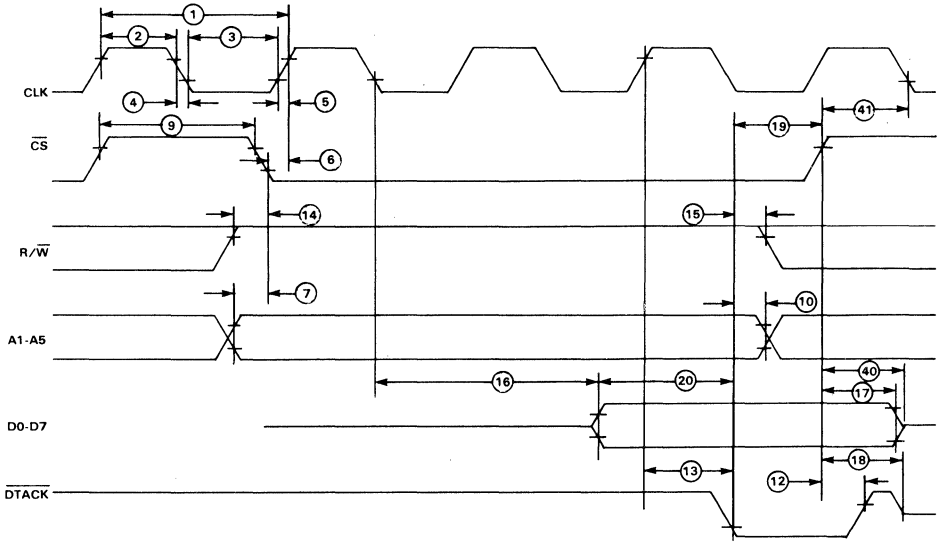
Figure 4



NOTE:
XTAL2 output test load is a cyrstal.

READ CYCLE

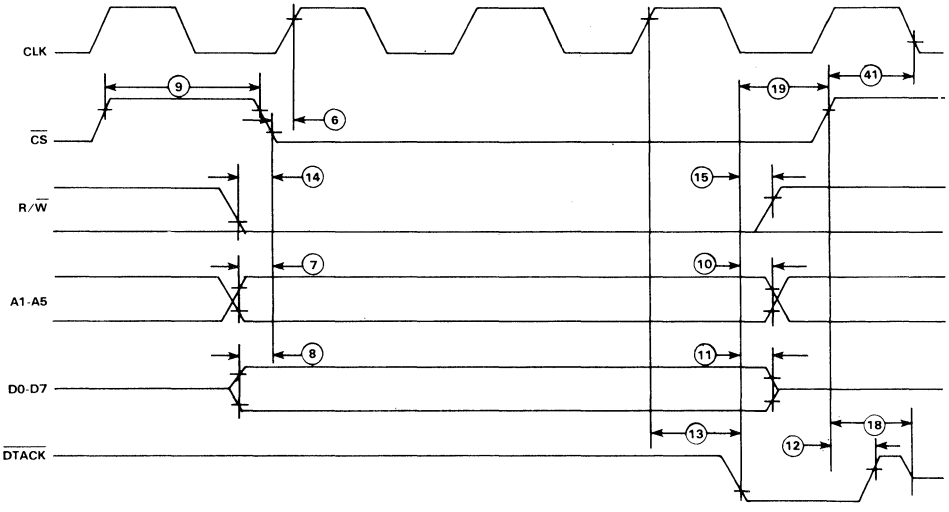
Figure 5



NOTE:
Waveform measurements for all inputs and outputs are specified at logic high = 2.0 volts, logic low = 0.8 volts.

WRITE CYCLE

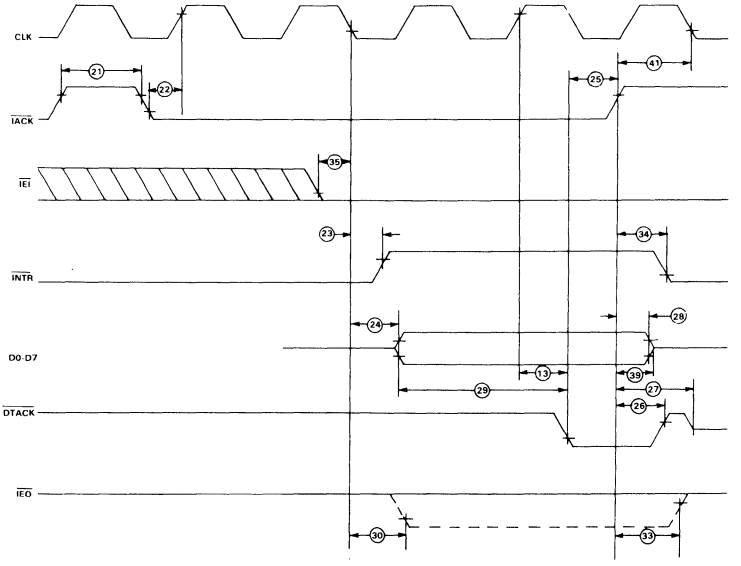
Figure 6



NOTE:
Waveform measurements for all inputs and outputs are specified at logic high = 2.0 volts, logic low = 0.8 volts.

INTERRUPT ACKNOWLEDGE CYCLE (IEI LOW)

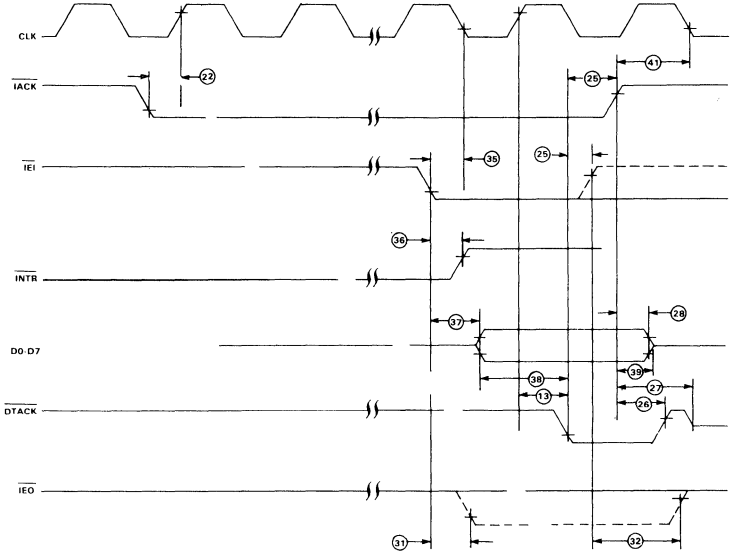
Figure 7



NOTE:
Waveform measurements for all inputs and outputs are specified at logic high = 2.0 volts, logic low = 0.8 volts.

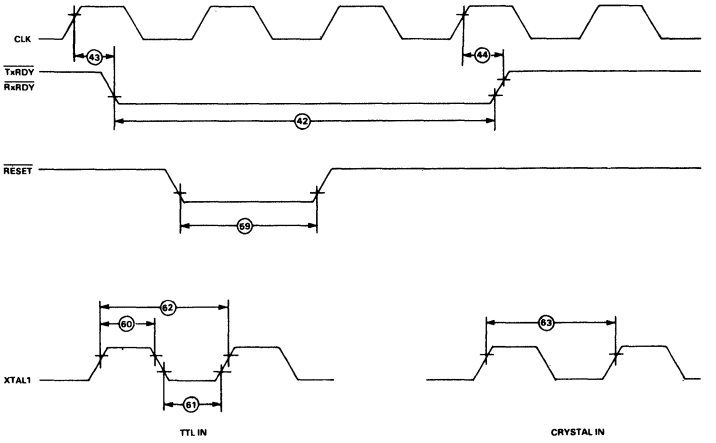
INTERRUPT ACKNOWLEDGE CYCLE (IEI HIGH)

Figure 8



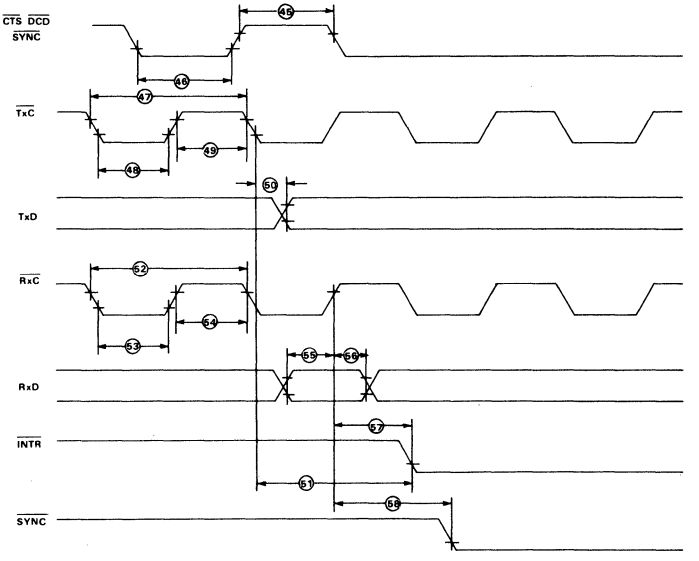
NOTE:
Waveform measurements for all inputs and outputs are specified at logic high = 2.0 volts, logic low = 0.8 volts.

DMA INTERFACE TIMING
Figure 9



NOTE:
Waveform measurements for all inputs and outputs are specified at logic high = 2.0 volts, logic low = 0.8 volts.

SERIAL INTERFACE TIMING
Figure 10



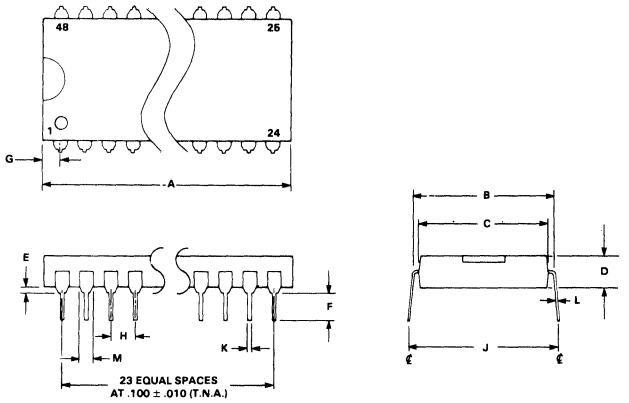
NOTE:
Waveform measurements for all inputs and outputs are specified at logic high = 2.0 volts, logic low = 0.8 volts.

MK68564 ORDERING INFORMATION

PART NO.	PACKAGE TYPE	MAX. CLOCK FREQUENCY	TEMPERATURE RANGE
MK68564N-03	Plastic	3.0 MHz	0° to 70°C
MK68564N-04	Plastic	4.0 MHz	0° to 70°C
MK68564N-05	Plastic	5.0 MHz	0° to 70°C

MK68564 PLASTIC PIN PACKAGE DRAWING

N SUFFIX
PLASTIC PACKAGE



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	61.468	62.738	2.420	2.470
B	14.986	16.256	.590	.640
C	13.462	13.97	.530	.550
D	3.556	4.064	.140	.160
E	0.381	1.524	.015	.060
F	3.048	3.81	.120	.150
G	1.524	2.286	.060	.090
H	2.286	2.794	.090	.110
J	15.24	17.78	.600	.700
K	0.381	0.533	.015	.021
L	0.203	0.305	.008	.012
M	1.143	1.778	.045	.070

THOMSON COMPONENTS MOSTEK

MK68901

MULTI-FUNCTION PERIPHERAL

MICROCOMPUTER COMPONENTS

FEATURES

- ☐ 8 Input/Output Pins
  - Individually programmable direction
  - Individual interrupt source capability
    - Programmable edge selection
- ☐ 16 Source interrupt controller
  - 8 Internal sources
  - 8 External sources
  - Individual source enable
  - Individual source masking
  - Programmable interrupt service modes
    - Polling
    - Vector generation
      - Optional In-service status
  - Daisy chaining capability
- ☐ Four timers with individually programmable prescaling
  - Two multimode timers
    - Delay mode
    - Pulse width measurement mode
    - Event counter mode
  - Two delay mode timers
  - Independent clock input
  - Time out output option
- ☐ Single channel USART
  - Full Duplex
  - Asynchronous to 65 kbps
  - Byte synchronous to 1 Mbps
  - Internal/external baud rate generation
  - DMA handshake signals
  - Modem control
  - Loop back mode
- ☐ 68000 Bus compatible
- ☐ 48 Pin DIP or 52 Pin PLCC

INTRODUCTION

The MK68901 MFP (Multi-Function Peripheral) is a combination of many of the necessary peripheral functions in a microprocessor system. Included are:

Eight parallel I/O lines

Interrupt controller for 16 sources

Four timers

Single channel full duplex USART

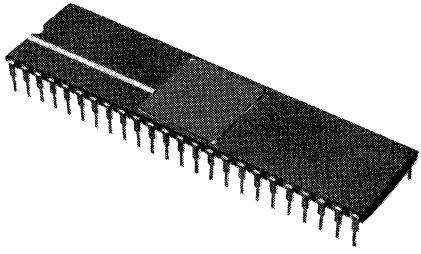


Figure 1. MK68901

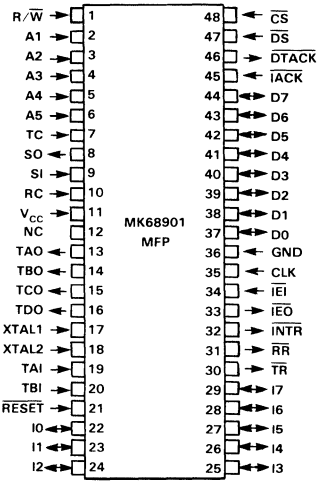


Figure 2. Device Pinout

The use of the MFP in a system can significantly reduce chip count, thereby reducing system cost. The MFP is completely 68000 bus compatible, and 24 directly addressable internal registers provide the necessary control and status interface to the programmer.

The MFP is a derivative of the MK3801 STI, a Z80 family peripheral.

PIN DESCRIPTION

GND: Ground

V_{CC}: +5 volts (± 5%)

$\overline{CS}$: Chip Select (input, active low). $\overline{CS}$ is used to select the MK68901 MFP for accesses to the internal registers. $\overline{CS}$ and $\overline{IACK}$ must not be asserted at the same time.

$\overline{DS}$: Data Strobe (input, active low.) $\overline{DS}$ is used as part of the chip select and interrupt acknowledge functions.

R/$\overline{W}$: Read/Write (input). R/$\overline{W}$ is the signal from the bus master indicating whether the current bus cycle is a Read (High) or Write (Low) cycle.

$\overline{DTACK}$: Data Transfer Acknowledge. (output, active low, tri-stateable). $\overline{DTACK}$ is used to signal the bus master that data is ready, or that data has been accepted by the MK68901 MFP.

A1-A5: Address Bus (inputs). The address bus is used to address one of the internal registers during a read or write cycle.

D0-D7: Data Bus (bi-directional, tri-stateable). The data bus is used to receive data from or transmit data to one of the internal registers during a read or write cycle. It is also used to pass a vector during an interrupt acknowledge cycle.

CLK: Clock (input). This input is used to provide the internal timing for the MK68901 MFP.

$\overline{RESET}$: Device reset. (input, active low). Reset disables the USART receiver and transmitter, stops all timers and forces the timer outputs low, disables all interrupt channels and clears any pending interrupts. The General Purpose Interrupt/I/O lines will be placed in the tri-state input mode. All internal registers (except the timer, USART data registers, and transmit status register) will be cleared.

$\overline{INTR}$: Interrupt Request (output, active low, open drain). $\overline{INTR}$ is asserted when the MK68901 MFP is requesting an interrupt. $\overline{INTR}$ is negated during an interrupt acknowledge cycle or by clearing the pending interrupt(s) through software.

$\overline{IACK}$: Interrupt Acknowledge (input, active low). $\overline{IACK}$ is used to signal the MK68901 MFP that the CPU is acknowledging an interrupt. $\overline{CS}$ and $\overline{IACK}$ must not be asserted at the same time.

$\overline{IEI}$: Interrupt Enable In (input, active low). $\overline{IEI}$ is used to signal the MK68901 MFP that no higher priority device is requesting interrupt service.

$\overline{IEO}$: Interrupt Enable Out (output, active low). $\overline{IEO}$ is used to signal lower priority peripherals that neither the MK68901 MFP nor another higher priority peripheral is requesting interrupt service.

I0-I7: General Purpose Interrupt I/O lines. These lines may be used as interrupt inputs and/or I/O lines. When used as interrupt inputs, their active edge is programmable. A data direction register is used to define which lines are to be Hi-Z inputs and which lines are to be push-pull TTL compatible outputs.

SO: Serial Output. This is the output of the USART transmitter.

SI: Serial Input. This is the input to the USART receiver.

RC: Receiver Clock. This input controls the serial bit rate of the USART receiver.

TC: Transmitter Clock. This input controls the serial bit rate of the USART transmitter.

$\overline{RR}$: Receiver Ready. (output, active low) DMA output for receiver, which reflects the status of Buffer Full in port number 15.

$\overline{TR}$: Transmitter Ready. (output, active low) DMA output for transmitter, which reflects the status of Buffer Empty in port number 16.

TAO, TBO, TCO, TDO: Timer Outputs. Each of the four timers has an output which can produce a square wave. The output will change states each timer cycle; thus one full period of the timer out signal is equal to two timer cycles. TAO or TBO can be reset (logic "0") by a write to TACR, or TBCR respectively.

XTAL1, XTAL2: Timer Clock inputs. A crystal can be connected between XTAL1 and XTAL2, or XTAL1 can be driven with a TTL level clock. When driving XTAL1 with a TTL level clock, XTAL2 must be allowed to float. When using a crystal, external capacitors are required. See Figure 33. All chip accesses are independent of the timer clock.

TAI, TBI: Timer A, B inputs. Used when running the timers in the event count or the pulse width measurement mode. The interrupt channels associated with 14 and 13 are used for TAI and TBI, respectively. Thus, when running a timer in the pulse width measurement mode, 14 or 13 can be used for I/O only.

INTERRUPTS

The General Purpose I/O-Interrupt Port (GPIP) provides eight I/O lines that may be operated either as inputs or outputs under software control. In addition, each line may generate an interrupt in either a positive going edge or a negative going edge of the input signal.

The GPIP has three associated registers. One allows the programmer to specify the Active Edge for each bit that will trigger an interrupt. Another register specifies the Data Direction (input or output) associated with each bit. The third register is the actual data I/O register used to input or output data to the port. These three registers are illustrated in Figure 5.

Address Port No.	Abbreviation	Register Name
0	GPIP	GENERAL PURPOSE I/O
1	AER	ACTIVE EDGE REGISTER
2	DDR	DATA DIRECTION REGISTER
3	IERA	INTERRUPT ENABLE REGISTER A
4	IERB	INTERRUPT ENABLE REGISTER B
5	IPRA	INTERRUPT PENDING REGISTER A
6	IPRB	INTERRUPT PENDING REGISTER B
7	ISRA	INTERRUPT IN-SERVICE REGISTER A
8	ISRB	INTERRUPT IN-SERVICE REGISTER B
9	IMRA	INTERRUPT MASK REGISTER A
A	IMRB	INTERRUPT MASK REGISTER B
B	VR	VECTOR REGISTER
C	TACR	TIMER A CONTROL REGISTER
D	TBCR	TIMER B CONTROL REGISTER
E	TCDCR	TIMERS C AND D CONTROL REGISTER
F	TADR	TIMER A DATA REGISTER
10	TBDR	TIMER B DATA REGISTER
11	TCDR	TIMER C DATA REGISTER
12	TDDR	TIMER D DATA REGISTER
13	SCR	SYNC CHARACTER REGISTER
14	UCR	USART CONTROL REGISTER
15	RSR	RECEIVER STATUS REGISTER
16	TSR	TRANSMITTER STATUS REGISTER
17	UDR	USART DATA REGISTER

Figure 4. Register Map

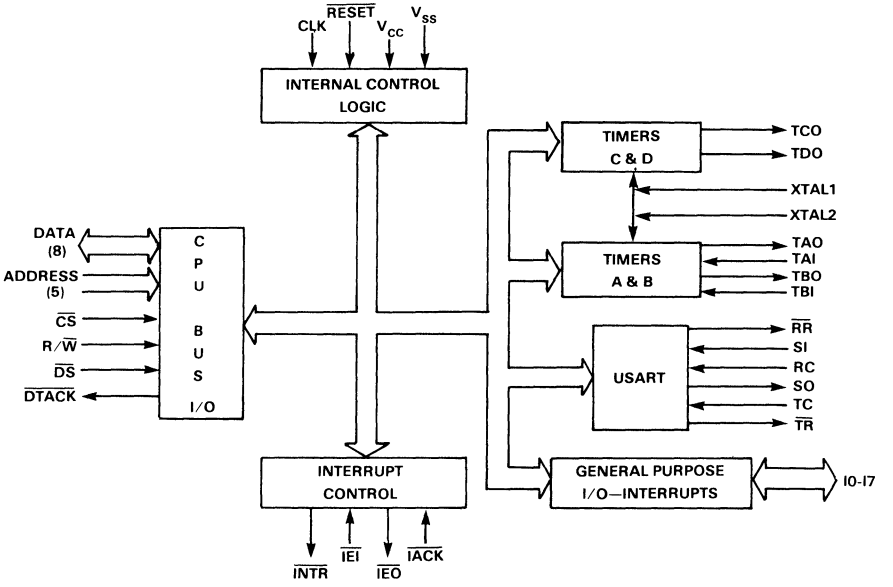


Figure 3. MK68901 Block Diagram

The Active Edge Register (AER) allows each of the General Purpose Interrupts to provide an interrupt on either a 1-0 transition or a 0-1 transition. Writing a zero to the appropriate bit of the AER causes the associated input to produce an interrupt on the 1-0 transition. The edge bit is simply one input to an exclusive-or gate, with the other input coming from the input buffer and the output going to a 1-0 transition detector. Thus, depending upon the state of the input, writing the AER can cause an interrupt-producing transition, which will cause an interrupt on the associated channel, if that channel is enabled. One would then normally configure the AER before enabling interrupts via IERA and IERB. Note: changing the edge bit, with the interrupt enabled, may cause an interrupt on that channel.

The Data Direction Register (DDR) is used to define I0-I7 as inputs or as outputs on a bit by bit basis. Writing a zero into a bit of the DDR causes the corresponding Interrupt-I/O pin to be a Hi-Z input. Writing a one into a bit of the DDR causes the corresponding pin to be configured as a push-pull output. When data is written into the GPIP, those pins defined as inputs will remain in the Hi-Z state while those pins defined as outputs will assume the state (high or low) of their corresponding bit in the GPIP. When the GPIP is read, the data read will come directly from the corresponding bit of the GPIP register for all pins defined as output, while the data read on all pins defined as inputs will come from the input buffers.

Each individual function in the MK68901 is provided with a unique interrupt vector that is presented to the system during the interrupt acknowledge cycle. The interrupt vector returned during the interrupt acknowledge cycle is shown in Figure 6, while the vector register is shown in Figure 7.

There are 16 vector addresses generated internally by the MK68901, one for each of the 16 interrupt channels.

The Interrupt Control Registers (Figure 8) provide control of interrupt processing for all I/O facilities of the MK68901. These registers allow the programmer to enable or disable any or all of the 16 interrupts, providing masking for any interrupt, and provide access to the pending and in-service status of the interrupt. Optional end-of-interrupt modes are available under software control. All the interrupts are prioritized as shown in Figure 9.

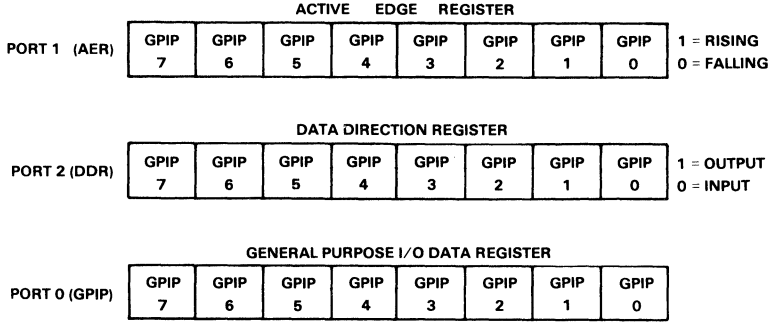


Figure 5. General Purpose I/O Registers

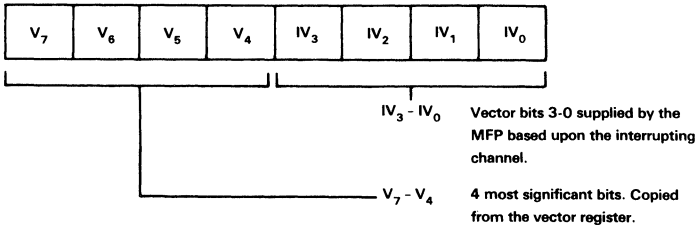


Figure 6. Interrupt Vector

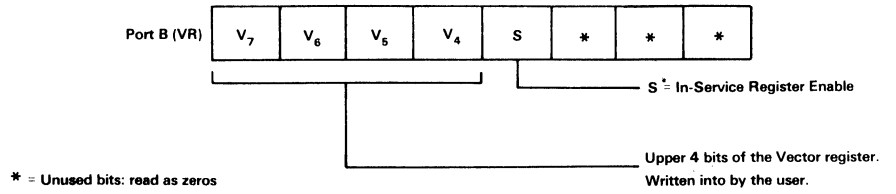


Figure 7. Vector Register

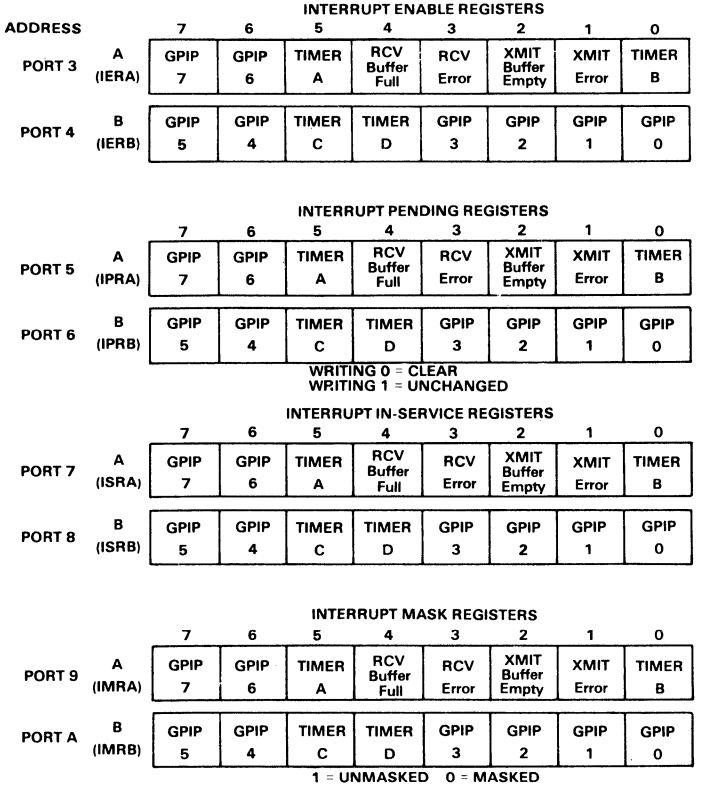


Figure 8. Interrupt Control Registers

Priority	Channel	Description
HIGHEST	1111	General Purpose Interrupt 7(I7)
	1110	General Purpose Interrupt 6(I6)
	1101	Timer A
	1100	Receive Buffer Full
	1011	Receive Error
	1010	Transmit Buffer Empty
	1001	Transmit Error
	1000	Timer B
	0111	General Purpose Interrupt 5(I5)
	0110	General Purpose Interrupt 4(I4)
	0101	Timer C
	0100	Timer D
	0011	General Purpose Interrupt 3(I3)
	0010	General Purpose Interrupt 2(I2)
	0001	General Purpose Interrupt 1(I1)
LOWEST	0000	General Purpose Interrupt 0(I0)

Figure 9. Interrupt Control Register Definitions

Interrupts may be either polled or vectored. Each channel may be individual enabled or disabled by writing a one or a zero in the appropriate bit of Interrupt Enable Registers (IERA, IERB - see Figure 8 for all registers in this section). When disabled, an interrupt channel is completely inactive. Any internal or external action which would normally produce an interrupt on that channel is ignored and any pending interrupt on that channel will be cleared by disabling that channel. Disabling an interrupt channel has no effect on the corresponding bit in Interrupt In-Service Registers (ISRA, ISRB); thus, if the In-service Registers are used and an interrupt is in service on that channel when the channel is disabled, it will remain in service until cleared in the normal manner. IERA and IERB are also readable.

When an interrupt is received on an enabled channel, its corresponding bit in the pending register will be set. When that channel is acknowledged it will pass its vector, and the corresponding bit in the Interrupt Pending Register (IPRA or IPRB) will be cleared. IPRA and IPRB are readable; thus by polling IPRA and IPRB, it can be determined whether a channel has a pending interrupt. IPRA and IPRB are also writeable and a pending interrupt can be cleared without going through the acknowledge sequence by writing a zero to the appropriate bit. This allows any one bit to be cleared, without altering any other bits, simply by writing all ones except for the bit position to be cleared to IPRA or IPRB. Thus a fully polled interrupt scheme is possible. Note: writing a one to IPRA, IPRB has no effect on the interrupt pending register.

The interrupt mask registers (IMRA and IMRB) may be used to block a channel from making an interrupt request. Writing a zero into the corresponding bit of the mask register will still allow the channel to receive an interrupt and latch it into its pending bit (if that channel is enabled), but will prevent that channel from making an interrupt request. If that channel is causing an interrupt request at the time the corresponding bit in the mask register is cleared, the request will cease. If no other channel is making a request, $\overline{INTR}$ will go inactive. If the mask bit is re-enabled, any pending interrupt is now free to resume its request unless blocked by a higher priority request for service. IMRA and IMRB are also readable. A conceptual circuit of an interrupt channel is shown in Figure 10.

There are two end-of-interrupt modes: the automatic end-of-interrupt mode and the software end-of-interrupt mode. The mode is selected by writing a one or a zero to the S bit of the Vector Register (VR). If the S bit of

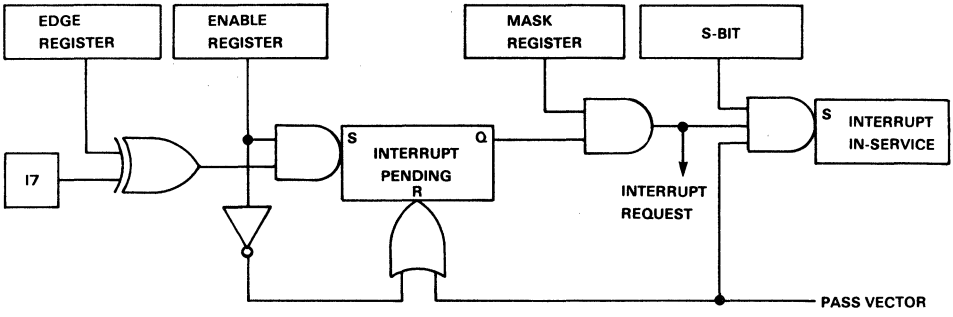


Figure 10. A Conceptual Circuit of an Interrupt Channel

the VR is a one, all channels operate in the software end-of-interrupt mode. If the S bit is a zero, all channels operate in the automatic end-of-interrupt mode, and a reset is held on all in-service bits. In the automatic end-of-interrupt mode, the pending bit is cleared when that channel passes its vector. At that point, no further history of that interrupt remains in the MK68901 MFP. In the software end-of-interrupt mode, the in-service bit is set and the pending bit is cleared when the channel passes its vector. With the in-service bit set, no lower priority channel is allowed to request an interrupt or to pass its vector during an acknowledge sequence; however, a lower priority channel may still receive an interrupt and latch it into the pending bit. A higher priority channel may still request an interrupt and be acknowledged. The in-service bit of a particular channel may be cleared by writing a zero to the corresponding bit in ISRA or ISRB. Typically, this will be done at the conclusion of the interrupt routine just before the return. Thus no lower priority channel will be allowed to request service until the higher priority channel is complete, while channels of still higher priority will be allowed to request service. While the in-service bit is set, a second interrupt on that channel may be received and latched into the pending bit, though no service request will be made in response to the second interrupt until the in-service bit is cleared. ISRA and ISRB may be read at any time. Only a zero may be written into any bit of ISRA and ISRB; thus the in-service bits may be cleared in software but cannot be set in software. This allows any one bit to be cleared, without altering any other bits, simply by writing all ones except for the bit position to be cleared to ISRA or ISRB, as with IPRA and IPRB.

Each interrupt channel responds with a discrete 8-bit vector when acknowledged. The upper four bits of the

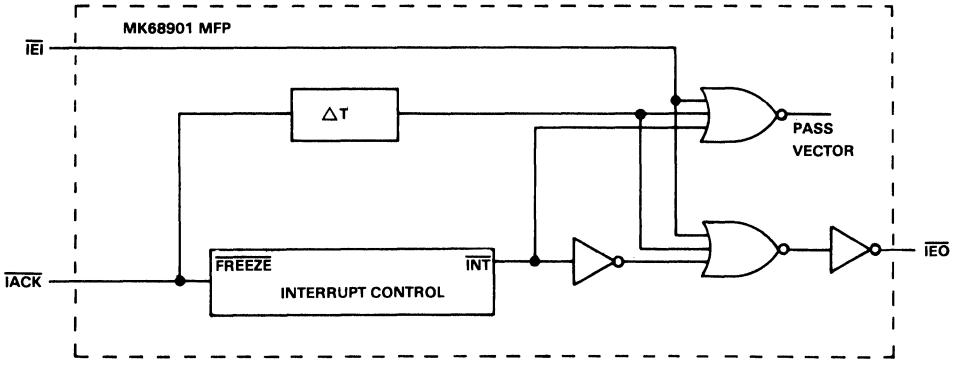


Figure 11a. A Conceptual Circuit of the MK68901 MFP Daisy Chaining

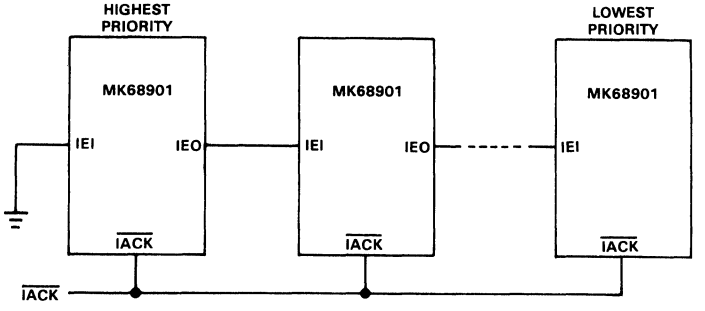


Figure 11b. Daisy Chaining

vector are set by writing the upper four bits of the VR. The four low order bits (Bit 3-Bit 0) are generated by the interrupting channel.

To acknowledge an interrupt, $\overline{IACK}$ goes low, the $\overline{IEI}$ input must go low (or be tied low) and the MK68901 MFP must have an acknowledgeable interrupt pending. The Daisy Chaining capability (Figure 11) requires that all parts in a chain have a common $\overline{IACK}$. When the common $\overline{IACK}$ goes low, all parts freeze and priortize interrupts in parallel. Then priority is passed down the chain, via $\overline{IEI}$ and $\overline{IEO}$, until a part which has a pending interrupt is reached. The part with the pending interrupt, passes a vector, does not propagate $\overline{IEO}$, and generates $\overline{DTACK}$.

Figure 9 describes the 16 prioritized interrupt channels. As shown, General Purpose Interrupt 7 has the highest priority, while General Purpose Interrupt 0 is assigned the lowest priority. Each of these channels may be reprioritized, in effect, by selectively masking interrupts under software control. The binary numbers under "channel" correspond to the modified bits IV3, IV2, IV1, and IV0, respectively, of the Interrupt Vector for each channel (see Figure 6).

Each channel has an enable bit contained in IERA or IERB, a pending latch contained in IPRA or IPRB, a mask bit contained in IMRA or IMRB, and an in-service latch contained in ISRA or ISRB. Additionally, the eight General Purpose Interrupts each have an edge bit contained in the Active Edge Register (AER), a bit to define the line as input or output contained in the Data Direction Register (DDR) and an I/O bit in the General Purpose Interrupt-I/O Port (GPIP).

TIMERS

There are four timers on the MK68901 MFP. Two of the timers (Timer A and Timer B) are full function timers which can perform the basic delay function and can also perform event counting, pulse width measurement, and waveform generation. The other two timers (Timer C and Timer D) are delay timers only. One or both of these timers can be used to supply the baud rate clocks for the USART. All timers are prescaler/counter timers with a common independent clock input (XTAL1, XTAL2). In addition, all timers have a time-out output function that toggles each time the timer times out.

The four timers are programmed via three Timer Control Registers and four Timer Data Registers. Timers A and B are controlled by the control registers TACR and TBCR, respectively (see Figure 12), and by the data registers TADR and TBDR (Figure 13). Timers C and D are controlled by the control register TCDCR (see Figure 14) and two data registers TCDR and TDDR. Bits in the control registers allow the selection of operational mode, prescale, and control while the data registers are used to read the timer or write into the time constant register. Timer A and B input pins, TAI and TBI, are used for the event and pulse width modes for timers A and B.

With the timer stopped, no counting can occur. The timer contents will remain unaltered while the timer is stopped (unless reloaded by writing the Timer Data Register), but any residual count in the prescaler will be lost.

In the delay mode, the prescaler is always active. A count pulse will be applied to the main timer unit each

Port C (TACR)	*	*	*	TIMER A RESET	AC_3	AC_2	AC_1	AC_0
Port D (TBCR)	*	*	*	TIMER B RESET	BC_3	BC_2	BC_1	BC_0

C_3	C_2	C_1	C_0	
0	0	0	0	Timer Stopped
0	0	0	1	Delay Mode, : 4 Prescale
0	0	1	0	Delay Mode, : 10 Prescale
0	0	1	1	Delay Mode, : 16 Prescale
0	1	0	0	Delay Mode, : 50 Prescale
0	1	0	1	Delay Mode, : 64 Prescale
0	1	1	0	Delay Mode, : 100 Prescale
0	1	1	1	Delay Mode, : 200 Prescale
1	0	0	0	Event Count Mode
1	0	0	1	Pulse Width Mode, : 4 Prescale
1	0	1	0	Pulse Width Mode, : 10 Prescale
1	0	1	1	Pulse Width Mode, : 16 Prescale
1	1	0	0	Pulse Width Mode, : 50 Prescale
1	1	0	1	Pulse Width Mode, : 64 Prescale
1	1	1	0	Pulse Width Mode, : 100 Prescale
1	1	1	1	Pulse Width Mode, : 200 Prescale

* Unused bits: read as zeros

Figure 12. Timer A and B Control Registers

time the prescribed number of timer clock cycles has elapsed. Thus, if the prescaler is programmed to divide by ten, a count pulse will be applied to the main counter every ten cycles of the timer clock.

Each time a count pulse is applied to the main counter, it will decrement its contents. The main counter is initially loaded by writing to the Timer Data Register. Each count pulse will cause the current count to decrement. When the timer has decremented down to "01", the next count pulse will not cause it to decrement to "00". Instead, the next count pulse will cause the timer to be reloaded from the Timer Data Register. Additionally, a "Time out" pulse will be produced. This Time Out pulse is coupled to the timer interrupt channel, and, if that channel is enabled, an interrupt will be produced. The Time Out pulse is also coupled to the timer output pin and will cause the pin to change states. The output will remain in this new state until the next Time Out pulse occurs. Thus the output will complete one full cycle for each two Time Out pulses.

If, for example, the prescaler were programmed to divide by ten, and the Timer Data Register were loaded with 100 (decimal), the main counter would decrement once for every ten cycles of the timer clock. A Time Out pulse will occur (hence an interrupt if that channel is enabled) every 1000 cycles of the timer clock, and the timer output will complete one full cycle every 2000 cycles of the timer clock.

The main counter is an 8-bit binary down counter. It may be read at any time by reading the Timer Data Register. The information read is the information last clocked into the timer read register when the $\overline{DS}$ pin had last gone high prior to the current read cycle. When written, data is loaded into the Timer Data Register, and the main counter, if the timer is stopped. If the Timer Data Register is written while the timer is running, the new word is not loaded into the timer until it counts through H"01". However, if the timer is written while it is counting through H"01", an indeterminate value will be written into the timer constant register. This may be circumvented by ensuring that the data register is not written when the count is H"01".

If the main counter is loaded with "01", a Time Out Pulse will occur every time the prescaler presents a count pulse to the main counter. If loaded with "00", a Time Out pulse will occur every 256 count pulses.

Changing the prescale value with the timer running can cause the first Time Out pulse to occur at an indeterminate time, (no less than one nor more than 200 timer

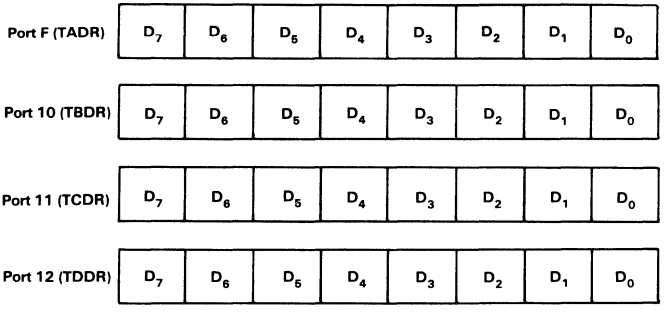
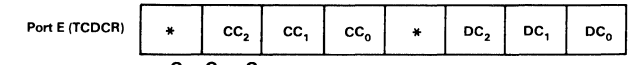


Figure 13. Timer Data Registers (A, B, C, and D)



C_2	C_1	C_0	
0	0	0	Timer Stopped
0	0	1	Delay Mode, ÷ 4 Prescale
0	1	0	Delay Mode, ÷ 10 Prescale
0	1	1	Delay Mode, ÷ 16 Prescale
1	0	0	Delay Mode, ÷ 50 Prescale
1	0	1	Delay Mode, ÷ 64 Prescale
1	1	0	Delay Mode, ÷ 100 Prescale
1	1	1	Delay Mode, ÷ 200 Prescale

* Unused bits: read as zeros

Figure 14. Timer C and D Register

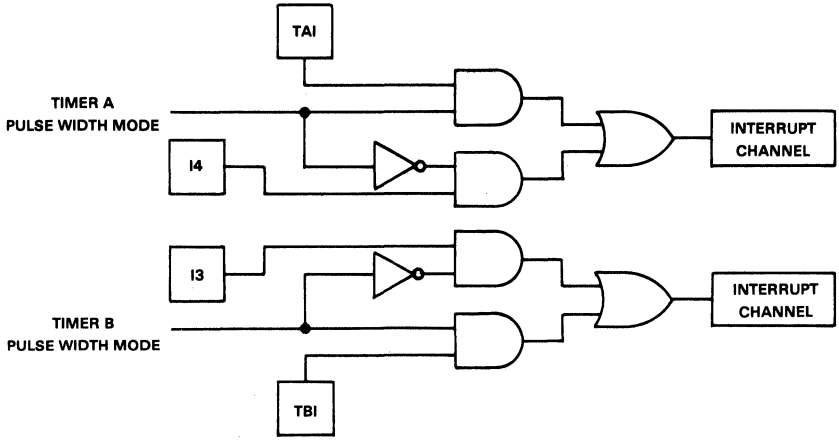


Figure 15. A Conceptual Circuit of the MFP Timers In the Pulse Width Measurement Mode

clock cycles times the number in the time constant register), but subsequent Time Out pulses will then occur at the correct interval.

In addition to the delay mode described above, Timers A and B can also function in the Pulse Width Measurement mode or in the Event Count mode. In either of these two modes, an auxiliary control signal is required. The auxiliary control input for Timer A is TAI, and for Timer B, TBI is used. The interrupt channels associated with I4 and I3 are used for TAI and TBI, respectively, in Pulse Width mode. See Figure 15.

The pulse width measurement mode functions much like the delay mode. However, in this mode, the auxiliary control signal on TAI or TBI acts as an enable to the timer. When the control signal on TAI or TBI is inactive, the timer will be stopped. When it is active, the prescaler and main counter are allowed to run. Thus the width of the active pulse on TAI or TBI is determined by the number of timer counts which occur while the pulse allows the timer to run. The active state of the signal on TAI or TBI is dependent upon the associated Interrupt Channel's edge bit (GPIP 4 for TAI and GPIP 3 for TBI; see Active Edge Register in Figure 5.) If the edge bit associated with the TAI or TBI input is a one, it will be active high; thus the timer will be allowed to run when the input is at a high level. If the edge bit is a zero, the TAI or TBI input will be active low. As previously stated, the interrupt channel (I3 or I4) associated with the input still functions when the timer is used in the pulse width measurement mode. However, if the timer is programmed for the pulse width measurement mode, the interrupt caused by transitions on the associated TAI or TBI input will occur on the opposite transition.

For example, if the edge bit associated with the TAI input (AER-GPIP 4) is a one, an interrupt would normally be generated on the 0-1 transition of the I4 input signal. If the timer asssociated with this input (Timer A) is placed in the pulse width measurement mode, the interrupt will occur on the 1-0 transition of the TAI signal instead. Because the edge bit (AER-GPIP 4) is a one, Timer A will be allowed to count while the input is high. When the TAI input makes the high to low transition, Timer A will stop, and it is at this point that the interrupt will occur (assuming that the channel is enabled). This allows the interrupt to signal the CPU that the pulse being measured has terminated; thus Timer A may now be read to determine the pulse width. (Again note that I3 and I4 may still be used for I/O when the timer is in the pulse width measurement mode.) If Timer A is reprogrammed for another mode, interrupts will again occur on the transition, as normally defined by the edge bit. Note that, like changing the edge bit, placing the timer into or taking it out of the pulse width mode can produce a transition on the signal to the interrupt channel and may cause an interrupt. If measuring consecutive pulses, it is obvious that one must read the contents of the timer and then reinitialize the main counter by writing to the timer data register. If the timer data register is written while the pulse is going to the active state, the write operation may result in an indeterminate value being written into the main counter. If the timer is written after the pulse goes active, the timer counts from the previous contents, and when it counts through H"01", the correct value is written into the timer. The

pulse width then includes counts from before the timer was reloaded.

In the event count mode, the prescaler is disabled. Each time the control input on TAI or TBI makes an active transition as defined by the associated Interrupt Channel's edge bit, a count pulse will be generated, and the main counter will decrement. In all other respects, the timer functions as previously described. Altering the edge bit while the timer is in the event count mode can produce a count pulse. The interrupt channel associated with the input (I3 for I4 for TAI) is allowed to function normally. To count transitions reliably, the input must remain in each state (1/O) for a length of time equal to four periods of the timer clock; thus signals of a frequency up to one fourth of the timer clock can be counted.

The manner in which the timer output pins toggle states has previously been described. All timer outputs will be forced low by a device RESET. The output associated with Timers A and B will toggle on each Time Out pulse regardless of the mode the timers are programmed to. In addition, the outputs from Timers A and B can be forced low at any time by writing a "1" to the reset location in TACR and TBCR, respectively. The output will be forced to the low state during the WRITE operation, and at the conclusion of the operation, the output will again be free to toggle each time a Time Out pulse occurs. This feature will allow waveform generation.

During reset, the Timer Data Registers and the main counters are not reset. Also, if using the reset option on Timers A or B, one must make sure to keep the other bits in the correct state so as not to affect the operation of Timers A and B.

USART

Serial Communication is provided by a full-duplex double-buffered USART, which is capable of either asynchronous or synchronous operation. Variable word length and start/stop bit configurations are available under software control for asynchronous operation. For synchronous operation, a Sync Word is provided to establish synchronization during receive operations. The Sync Word will also be repeatedly transmitted when no other data is available for transmission. Moreover, the MK68901 allows stripping of all Sync Words received in synchronous operation. The handshake control lines RR (Receiver Ready) and TR (Transmitter Ready) allow DMA operation. Separate receive and transmit clocks are available, and separate receive and transmit status and data bytes allow independent operation of the transmit and receive sections.

The USART is provided with three Control/Status Registers and a Data Register. The USART Data Register form is illustrated in Figure 16. The programmer may specify operational parameters for the USART via the Control Register, as shown in Figure 17. Status of both the Receiver and Transmitter sections is accessed by means of the two Status Registers, as shown in Figures 18 and 19. Data written to the Data Register is passed to the transmitter, while reading the Data Register will access data received by the USART.

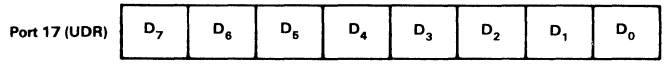
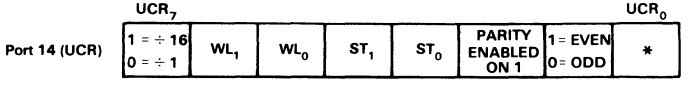


Figure 16. USART Data Register



* Unused bits; read as zero

Figure 17. USART Control Register (UCR)

÷16/÷1: When this bit is zero, data will be clocked into and out of the receiver and transmitter at the frequency of their respective clocks. When this bit is loaded with a one, data will be clocked into and out of the receiver and transmitter at one sixteenth the frequency of their respective clocks. Additionally, when placed in the divide by sixteen mode, the receiver data transition resynchronization logic will be enabled.

WL0-WL1: Word Length Control. These two bits set the length of the data word (exclusive of start bits, stop bits, and parity bits as follows:

WL1	WL0	Word Length
0	0	8 bits
0	1	7 bits
1	0	6 bits
1	1	5 bits

ST0-ST1: Start/Stop bit control (format control). These two bits set the format as follows:

ST1	ST0	Start Bits	Stop Bits	Format
0	0	0	0	SYNC
0	1	1	1	ASYNC
†1	0	1	1½	ASYNC
1	1	1	2	ASYNC

†NOTE ÷ 16 only

PARITY: Parity Enabled. When set ("1"), parity will be checked by the receiver, parity will be calculated, and a parity bit will be inserted by the transmitter. When cleared ("0"), no parity check will be made and no parity bit will be inserted for transmission.

For a word length of 8 the MFP calculates the parity and appends it when transmitting a sync character. For shorter lengths, the parity must be stored in the Sync Character Register (SCR) along with the sync character.

E/O: Even-Odd. When set ("1"), even parity will be used if parity is enabled. When cleared ("0"), odd parity will be used if parity is enabled.

Note that the synchronous or asynchronous format may be selected independently of a ÷1 or ÷16 clock. Thus it is possible to clock data synchronously into the device but still use start and stop bits. In this mode, all normal asynchronous format features still apply. Data will be shifted in after a start bit is encountered, and a stop bit will be checked to determine proper framing. If a transmit underrun condition occurs, the output will be placed in a marking state, etc. It is conversely possible to clock data in asynchronously using a synchronous format. There is data transition detection logic built into the receive clock circuitry which will re-synchronize the internal shift clock on each data transition so that, with sufficiently frequent data transitions, start bits are not required. In this mode, all other common synchronous features function normally. This re-synchronization logic is only active in ÷16 clock mode.

RECEIVER

The receiver section of the USART is configured by the UCR as previously described. The status of the receiver can be determined by reading and writing to the Receiver Status Register (RSR). The RSR is configured as follows:

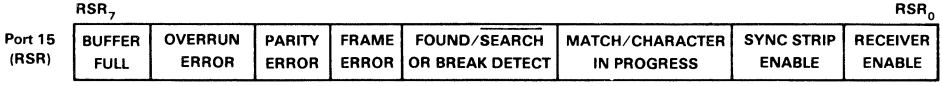


Figure 18. Receiver Status Register (RSR)

BF: Buffer Full. This bit is set when the incoming word is transferred to the receive buffer. The bit is cleared when the receive buffer is read by reading the UDR. This bit of the RSR is read only.

OE: Overrun Error. This flag is set if the incoming word is completely received and due to be transferred to the receive buffer, but the last word in the receive buffer has not yet been read. When this condition occurs, the word in the receive buffer is not overwritten by the new word. Note that the status flags always reflect the status of the data word currently in the receive buffer. As such, the OE flag is not actually set until

the good word currently in the buffer has been read. The interrupt associated with this error will also not be generated until the old word in the receive buffer has been read.

OE flag is cleared by reading the receiver status register, and new data words cannot be shifted to the receive buffer until this is done.

PE: Parity Error. This flag is set if the word received has a parity error. The flag is set when the received word is transferred from the shift register to the receive buffer if the error condition exists. The flag is cleared when the next word which does not have a parity error is transferred to the receive buffer.

FE: Frame Error. This flag only applies to the asynchronous format. A frame error is defined as a non-zero data word which is not followed by a stop bit. Like the PE flag, the FE flag is set or cleared when a word is transferred to the receive buffer.

F/$\overline{S}$: Found/$\overline{Search}$. This combination control bit and flag bit is only used with the synchronous format. It can be set or cleared by writing to this bit of the RSR. When this bit is cleared, the receiver is placed in the search mode. In this mode, a bit by bit comparison of the incoming data to the character in the Sync Character Register (SCR) is made. The word length counter is disabled. When a match is found, this bit will be set automatically, and the word length counter will start as sync has not been achieved. An interrupt will be generated on the receive error channel when the match occurs. The word just shifted in will, of necessity, be equal to the sync character, and it will not be transferred to the receive buffer.

B: Break. This flag is used only when the asynchronous format is selected. This flag will be set when an all zero data word, followed by no stop bit, is received. The flag will stay set until both a non-zero bit is received and the RSR has been read at least once since the flag was set. Break indication will not occur if the receive buffer is full.

M/CIP: Match/Character in Progress. If the synchronous format is selected, this flag is the Match flag. It will be set each time the word transferred to the receive buffer matches the sync character. It will be reset each time the word transferred to the receive buffer does not match the sync character. If the asynchronous format is selected, this flag represents Character in Progress. It will be set upon a start bit detect and cleared at the end of the word.

SS: Sync Strip Enable. If this bit is set to a one, data words that match the sync character will not be loaded into the receive buffer, and no buffer full signal will be generated.

RE: Receiver Enable. This control bit is used to enable or disable the receiver. If a zero is written to this bit of the RSR, the receiver will turn off immediately. All flags including the F/$\overline{S}$ bit will be cleared. If a one is written to this bit, normal receiver operation is enabled. The receive clock has to be running before the receiver is enabled.

There are two interrupt channels associated with the receiver. One channel is used for the normal Buffer Full condition, while the other channel is used whenever an error condition occurs. Only one interrupt is generated per word received, but dedicating two channels allows separate vectors: one for the normal condition, and one for an error condition. If the error channel is disabled, an interrupt will be generated via the Buffer Full Channel, whether the word received is normal or in error. Those conditions which produce an interrupt via the error channel are: Overrun, Parity Error, Frame Error, Sync Found, and Break. If a received word has an error associated with it, and the error interrupt channel is enabled, an interrupt will occur on the error channel only.

Each time a word is transferred into the receive buffer, a corresponding set of flags is latched into the RSR. No flags (except CIP) are allowed to change until the data word has been read from the receive buffer. Reading the receive buffer allows a new data word to be transferred to the receive buffer when it is received. Thus one should first read the RSR then read the receive buffer (UDR) to ensure that the flags just read match the data word just read. If done in the reverse order, it is possible that subsequent to reading the data word from the receive buffer, but prior to reading the RSR, a new word may be received and transferred to the receive buffer and, with it, its associated flags latched into the RSR. Thus, when the RSR is read, those flags may actually correspond to a different data word. It is good practice, also to read the RSR prior to a data read as, when an overrun error occurs, the receiver will not assemble new characters until the RSR has been read.

As previously stated, when overrun occurs, the OE flag will not be set and the associated interrupt will not be

generated until the receive buffer has been read. If a break occurs, and the receive buffer has not yet been read, only the B flag will be set (OE will not be set). Again, this flag will not be set until the last valid word has been read from the receive buffer. If the break condition ends and another whole data word is received before the receive buffer is read, both the B and OE flags will be set once the receive buffer is read.

If a break occurs while the OE flag is set, the B flag will also be set.

A break generates an interrupt when the condition occurs and again when the condition ends. If the break condition ends before it is acknowledged by reading the RSR, the receiver error interrupt indicating end of break will be generated once the RSR is read.

Anytime the asynchronous format is selected, start bit detection is enabled. New data is not shifted into the shift register until a zero bit is detected. If a ÷16 clock is selected, along with the asynchronous format, false start bit detection is also enabled. Any transition has to be stable for 3 positive going edges of the receive clock to be called a valid transition. For a start bit to be good, a valid 0-1 transition must not occur for 8 positive clock transitions after the initial valid 1-0 transition.

After a good start bit has been detected, valid transitions in the data are checked for continously. When a valid transition is detected, the counter is forced to state zero, and no more transition checking is started until state four. At state eight, the "previous state" of the transition checking logic is clocked into the receiver.

As a result of this resynchronization logic, it is possible to run with asynchronous clocks without start and stop bits if there are sufficient valid transitions in the data stream. This logic also makes the unit more tolerant of clock skew for normal asynchronous communications than a device which employs only start bit synchronization.

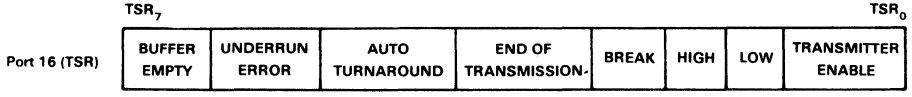


Figure 19. Transmitter Status Register (TSR)

TRANSMITTER

The transmitter section of the USART is configured as to format, word length, etc. by the UCR, as previously described. The status of the transmitter can be determined by reading or writing the Transmitter Status Register (TSR). The TSR is configured as follows:

BE: Buffer Empty. This status bit is set when the word in the transmit buffer is transferred to the output shift register and thus the transmit buffer may be reloaded with the next data word. The flag is cleared when the transmit buffer is reloaded. The transmit buffer is loaded by writing to the UDR.

UE: This bit is set when the last word has been shifted out of the transmit shift register before a new word has been loaded into the transmit buffer. It is not necessary to clear this bit before loading the UDR.

This bit may be cleared by either reading the TSR or by disabling the transmitter. After the setting of the UE bit, one full transmitter clock cycle is required before this bit can be cleared by a read. The timing in some systems may allow a read of the TSR before the required clock cycle has been completed. This would result in the UE bit not being cleared until the following read. To avoid this problem, a dummy read of the TSR should be performed at the end of the UE service routine.

Only one underrun error may be generated between loads of the UDR regardless of the number of transmitter clock cycles between UDR loads.

AT: This bit causes the receiver to be enabled at the end of the transmission of the last word in the transmitter if the transmitter has been disabled.

END: End of Transmission. When the transmitter is turned off with a character still in the output shift register, transmission will continue until that character is shifted out. Once it has cleared the output register, the END bit will be set. If no character is being transmitted when the transmitter is disabled, the transmitter will stop at the next rising edge of the internal shift clock, and END will immediately be set. The END bit is cleared by re-enabling the transmitter.

B: Break. This control bit will cause a break to be transmitted. When a "1" is written to the B bit of the TSR, a break will be transmitted upon completion of the character (if any) currently being transmitted. A break will continue to be transmitted until the B bit is cleared by writing a "0" to this bit of the TSR. At that time, normal transmission will resume. The B bit has no function in the synchronous format. Setting the "B" bit to a one keeps the "BE" bit from being set to a one. So, if there were a word in the buffer at the start of break, it would remain there until the end of break, at which time it would be transmitted (if the transmitter is still enabled). If the buffer were not full at the start of break, it could be written at any time during the break. If the buffer is empty at the end of break, the underrun flag will be set (unless the transmitter is disabled).

The BREAK bit cannot be set until the transmitter has been enabled and the transmitter has had sufficient time (one clock cycle) to perform the internal reset and initialization functions.

H,L: High and Low. These two control bits are used to configure the transmitter output, when the transmitter is disabled, as follows:

H	L	Output State
0	0	Hi-Z
0	1	Low ("0")
1	0	High
1	1	Loop-Connects transmitter output to receiver input, and TC to Receiver Clock (RC and SI are not used; they are bypassed internally). In loop back mode, transmitter output goes high when disabled.

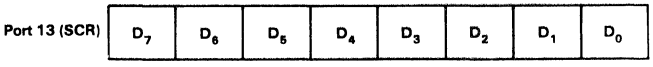


Figure 20. SYNC Character Register

Altering these two bits after Transmitter Enable (XE) is set will alter the output state until END is false. These bits should be set prior to enabling the transmitter. The state of these bits determine the state of the first transmitted character after the transmitter is enabled. If the high impedance mode was selected prior to the transmitter being enabled, the first bit transmitted is indeterminate.

XE: Transmitter Enable. This control bit is used to enable or disable the transmitter. When set, the transmitter is enabled. When cleared, the transmitter will be disabled. If disabled, any word currently in the output register will continue to be transmitted until finished. If a break is being transmitted when XE is cleared, the transmitter will turn off at the end of the break character boundary, and no end of break stop bit is transmitted. The transmit clock must be running before the transmitter is enabled. A "one" bit always precedes the first word out of the transmitter after the transmitter is enabled. There is a delay between the time the transmitter enable bit is written and when the transmitter reset goes low; therefore, the H & L bits should be written with the desired state prior to enabling the transmitter.

Like the receiver section, there are two separate interrupt channels associated with the transmitter. The buffer Empty condition causes an interrupt via one channel, while the Underrun and END conditions will cause an interrupt via the second channel. When underrun occurs in the synchronous format, the character in the SCR will be transmitted until a new word is loaded into the transmit buffer. In the asynchronous format, a "Mark" will be continuously transmitted when underrun occurs.

The transmit buffer can be loaded prior to enabling the transmitter. When the transmitter is disabled, any character currently in the process of being transmitted will continue to conclusion, but any character in the transmit buffer will not be transmitted and will remain in the buffer. Thus no buffer empty interrupt will occur nor will the BE flag be set. If the buffer were already empty, the

BE flag would be set and would remain set. When the transmitter is disabled with a character in the output register but with no character in the transmit buffer, an Underrun Error will not occur when the character in progress concludes.

Often it is necessary to send a break for some particular period. To aid in timing a break transmission, a transmit error interrupt will be generated at every normal character boundary time during a break transmission. The status register information is unaffected by this error condition interrupt. It should be noted that an underrun error, if present, must be cleared from the TSR, and the interrupt pending register must be cleared of pending transmitter errors at the beginning of the break transmission or no interrupts will be generated at the character boundary time.

If the synchronous format is selected, the sync character should be loaded into the Sync Character Register (SCR) as shown in Figure 20. This character is compared to the received serial data during a Search, and will be continuously transmitted during an underrun condition.

All flags in the RSR or TSR will continue to function as described whether their associated interrupt channel is disabled or enabled. All interrupt channels are edge triggered and, in many cases, it is the actual output of a flag bit or flag bits which is coupled to the interrupt channel. Thus, if a normal interrupt producing condition occurs while the interrupt channel is disabled, no interrupt would be produced even if the channel was subsequently enabled, because a transition did not occur while the interrupt channel was enabled. That particular flag bit would have to occur a second time before another "edge" was produced, causing an interrupt to be generated.

Error conditions in the USART are determined by monitoring the Receive Status Register and the Transmitter Status Register. These error conditions are only valid for each word boundary and are not latched. When executing block transfers of data, it is necessary to save any errors so that they can be checked at the end of a block. In order to save error conditions during data transfer, the MK68901 MFP interrupt controller may be used by enabling error interrupts for the desired channel (Receive error or Transmit error) and by masking these bits off. Once the transfer is complete, the Interrupt Pending Register can be polled, to determine the presence of a pending error interrupt, and therefore an error.

Unused bits in the sync character register are zeroed out; therefore, word length should be set up prior to writing the sync word in some cases. Sync word length is the word length plus one when parity is enabled. The user has to determine the parity of the sync word when the word length is not 8 bits. The MK68901 MFP does not add a parity bit to the sync word if the word length is less than 8 bits. The extra bit in the sync word is transmitted as the parity bit. With a word length of eight, and parity selected, the parity bit for the sync word is computed and added on by the MK68901 MFP.

$\overline{RR}$ RECEIVER READY

$\overline{RR}$ is asserted when the Buffer Full bit is set in the RSR unless a parity error or frame error is detected by the receiver.

$\overline{TR}$ TRANSMITTER READY

$\overline{TR}$ is asserted when the Buffer Empty bit is set in the TSR unless a break is currently being transmitted.

REGISTER ACCESSES

All register accesses are dependent on CLK as shown in the timing diagrams. To read a register, $\overline{CS}$ and $\overline{DS}$ must be asserted, and R/$\overline{W}$ must be high. The internal read control signal is essentially the combination of $\overline{CS}$, $\overline{DS}$, and RD/$\overline{WR}$. Thus, the read operation will begin when $\overline{CS}$ and $\overline{DS}$ go active and will end when either $\overline{CS}$ or $\overline{DS}$ goes inactive. The address bus must be stable prior to the start of the operation and must remain stable until the end of the operation. Unless a read operation or interrupt acknowledge cycle is in progress the data bus (D_0-D_7) will remain in the tri-state condition.

To write a register, $\overline{CS}$ and $\overline{DS}$ must be asserted and R/$\overline{W}$ must be low. The address must be stable prior to the start of the operation and must remain stable until the end of the operation. After the MK68901 asserts $\overline{DTACK}$, the CPU negates $\overline{DS}$. At this time, the MFP latches the data bus and writes the contents into the appropriate register. Also when $\overline{DS}$ is negated, the MFP rescinds $\overline{DTACK}$.

For an interrupt acknowledge, the operation starts when $\overline{IACK}$ goes low, and ends when $\overline{IACK}$ goes high. The data bus is tri-stated when either $\overline{IACK}$ or $\overline{DS}$ goes high.

When $\overline{CS}$ or $\overline{IACK}$ are asserted the MFP starts an internal cycle. $\overline{DS}$ is needed to enable the address and data buffers. It is recommended that $\overline{CS}$ and $\overline{IACK}$ be gated by $\overline{DS}$ so that $\overline{DS}$ is always present whenever an MFP bus cycle starts.

MK68901 ELECTRICAL SPECIFICATIONS - PRELIMINARY

ABSOLUTE MAXIMUM RATINGS

Temperature Under Bias ..... −25°C to +100°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with Respect to Ground ..... −0.3 V to +7 V
Power Dissipation ..... 1.5 W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

D.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = +5 V ± 5% unless otherwise specified.

SYM	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
V_{IH}	Input High Voltage	2.0	V_{CC} + .3	V	
V_{IL}	Input Low Voltage	−0.3	0.8	V	
V_{OH}	Output High Voltage (except $\overline{DTACK}$)	2.4		V	I_{OH} = −120μA
V_{OL}	Output Low Voltage (except $\overline{DTACK}$)		0.5	V	I_{OL} = 2.0 mA
I_{LL}	Power Supply Current		180	mA	Outputs Open
I_{LI}	Input Leakage Current		±10	μA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float		10	μA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float		−10	μA	V_{OUT} = 0.5 V
I_{OH}	$\overline{DTACK}$ output source current		−400	μA	V_{OUT} = 2.4
I_{OL}	$\overline{DTACK}$ output sink current		5.3	mA	V_{OUT} = 0.5

All voltages are referenced to ground

CAPACITANCE

T_A = 25°C, f = 1 MHz unmeasured pins returned to ground.

SYM	PARAMETER	MAX	UNIT	TEST CONDITION
C_{IN}	Input Capacitance	10	pF	Unmeasured pins returned to ground
C_{OUT}	Tri-state Output Capacitance	10	pF	

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 5%, GND = 0 Vdc, T_A = 0°C to 70°C)

NUM	CHARACTERISTIC	MK68901-0		MK68901-5		UNIT	FIG	NOTE
		MIN	MAX	MIN	MAX			
1	$\overline{CS}$, $\overline{DS}$ Width High	50		35		ns	21,22	5
2	R/$\overline{W}$, A1-A5 Valid to falling $\overline{CS}$ (Setup)	0		0		ns	21,22	
3	Data Valid Prior to Falling CLK	280		0		ns	22	
4	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (Setup)	50		45		ns	21-24	3
5	CLK Low to $\overline{DTACK}$ Low		220		180	ns	21,22	
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ high		60		55	ns	21-24	
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state		100		95	ns	21-24	
8	$\overline{DTACK}$ Low to Data Invalid (Hold Time)	0		0		ns	22	
9	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to Data Tri-state		50		50	ns	21,23,24	
10	$\overline{CS}$ or $\overline{DS}$ High to R/$\overline{W}$, A1-A5 Invalid (Hold Time)	0		0		ns	21,22	
11	Data Valid from $\overline{CS}$ Low		310		260	ns	21	3,6
12	Read Data Valid to $\overline{DTACK}$ Low (Setup Time)	50		50		ns	21	
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (Hold Time)	0		0		ns	21-23	
14	$\overline{IEI}$ low to falling $\overline{CLK}$ (Setup)	50		50		ns	23,24	
15	$\overline{IEO}$ Valid from Clock Low (Delay)		180		180	ns	23	1
16	Data Valid From Clock Low (Delay)		300		300	ns	23	
17	$\overline{IEO}$ Invalid from $\overline{IACK}$ High (Delay)		150		150	ns	23,24	
18	$\overline{DTACK}$ Low from Clock High (Delay)		180		165	ns	23,24	
19	$\overline{IEO}$ Valid from $\overline{IEI}$ Low (Delay)		100		100	ns	24	1
20	Data Valid from $\overline{IEI}$ Low (Delay)		220		220	ns	24	
21	Clock Cycle Time	250	1000	200	1000	ns	21	
22	Clock Width Low	110		90		ns	21	
23	Clock Width High	110		90		ns	21	
24	$\overline{CS}$, $\overline{IACK}$ Inactive to Rising Clock (Setup)	100		80		ns	21-23	4,5
25	I/O Minimum Active Pulse Width	100		100		ns	25	
26	$\overline{IACK}$ width High	2		2		T_{CLK}	23-24	2
27	I/O Data valid from Rising $\overline{CS}$ or $\overline{DS}$		450		450	ns	26	
28	Receiver Ready Delay from Rising RC		600		600	ns	27	
29	Transmitter Ready Delay from Rising TC		600		600	ns	28	
30	Timer Output Low from Rising Edge of $\overline{CS}$ or $\overline{DS}$ (A & B) (Reset T_{OUT})		450		450	ns	29	7
31	T_{OUT} Valid from Internal Timeout		2 t_{CLK} +300		2 t_{CLK} +300	ns	29	2
32	Timer Clock Low Time	110		90		ns	29	

AC ELECTRICAL CHARACTERISTICS (Continued) (V_{CC} = 5.0 Vdc ± 5%, GND = 0 Vdc, T_A = 0°C to 70°C)

NUM	CHARACTERISTIC	MK68901-0 MIN	MK68901-0 MAX	MK68901-5 MIN	MK68901-5 MAX	UNIT	FIG	NOTE
33	Timer Clock High Time	110		90		ns	29	
34	Timer Clock Cycle Time	250	1000	200	1000	ns	29	
35	$\overline{\text{RESET}}$ Low Time	2		1.8		μs	30	
36	Delay to Falling $\overline{\text{INTR}}$ from External Interrupt Active Transition		380		380	ns	25	
37	Transmitter Internal Interrupt Delay from Falling Edge of TC		550		550	ns	28	
38	Receiver Buffer Full Interrupt Transition Delay from Rising Edge of RC		800		800	ns	27	
39	Receiver Error Interrupt Transition Delay from Falling Edge of RC		800		800	ns	27	
40	Serial In Set Up Time to Rising Edge of RC (Divide by one only)	80		70		ns	27	
41	Data Hold Time from rising edge of RC (Divide by one only)	350		325		ns	27	
42	Serial Output Data Valid from Falling Edge of TC (÷1)		440		420	ns	28	
43	Transmitter Clock Low Time	500		450		ns	28	
44	Transmitter Clock High Time	500		450		ns	28	
45	Transmitter Clock Cycle Time	1.05	∞	0.95	∞	μs	28	
46	Receiver Clock Low Time	500		450		ns	27	
47	Receiver Clock High Time	500		450		ns	27	
48	Receiver Clock Cycle Time	1.05	∞	0.95	∞	μs	27	
49	$\overline{\text{CS}}$, $\overline{\text{IACK}}$, $\overline{\text{DS}}$ Width Low		80		80	T_{CLK}	29	2
50	Serial Output Data Valid from Falling Edge of TC (÷16)		490		370	ns	28	

NOTES:

1. $\overline{\text{IEO}}$ only goes low if no acknowledgeable interrupt is pending. If $\overline{\text{IEO}}$ goes low, $\overline{\text{DTACK}}$ and the data bus remain tri-stated.
2. T_{CLK} refers to the clock applied to the MFP CLK input pin. t_{CLK} refers to the timer clock signal, regardless of whether that signal comes from the XTAL 1/XTAL2 crystal clock inputs or the TAI or TBI timer inputs.
3. If the setup time is not met, $\overline{\text{CS}}$ or $\overline{\text{IACK}}$ will not be recognized until the next falling CLK.
4. If this setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.
5. $\overline{\text{CS}}$ is latched internally, therefore if spec's 1 and 24 are met then $\overline{\text{CS}}$ may be reasserted before the rising clock and still terminate the current bus cycle. The new bus cycle will be delayed by the MK68901 until all appropriate internal operations have completed.
6. Although $\overline{\text{CS}}$ and $\overline{\text{DTACK}}$ are synchronized with the clock, the data out during a read cycle is asynchronous to the clock, relying only on $\overline{\text{CS}}$ for timing.
7. Spec. 30 applies to timer outputs TAO and TBO only.

TIMER A.C. CHARACTERISTICS

Definitions:

Error = Indicated Time Value - Actual Time Value

tpsc = t_{CLK}x Prescale Value

Internal Timer Mode

Single Interval Error (free running) (Note 2)	±100 ns
Cumulative Internal Error	0
Error Between Two Timer Reads	±(tpsc + 4 t_{CLK})
Start Timer to Stop Timer Error	+(2 t_{CLK} + 100 ns) to − (tpsc + 6t_{CLK} + 100 ns)
Start Timer to Read Timer Error	+0 to − (tpsc + 6t_{CLK} + 400 ns)
Start Timer to Interrupt Request Error (Note 3)	−2 t_{CLK} to − (4t_{CLK} + 800 ns)

Pulse Width Measurement Mode

Measurement Accuracy (Note 1)	+2 t_{CLK} to − (tpsc + 4t_{CLK})
Minimum Pulse Width	4t_{CLK}

Event Counter Mode

Minimum Active Time of TAI, TBI	4t_{CLK}
Minimum Inactive Time of TAI, TBI	4t_{CLK}

NOTES:
1. Error may be cumulative if repetitively performed.
2. Error with respect to T_{OUT} or INT if note 3 is true.
3. Assuming it is possible for the timer to make an interrupt request immediately.

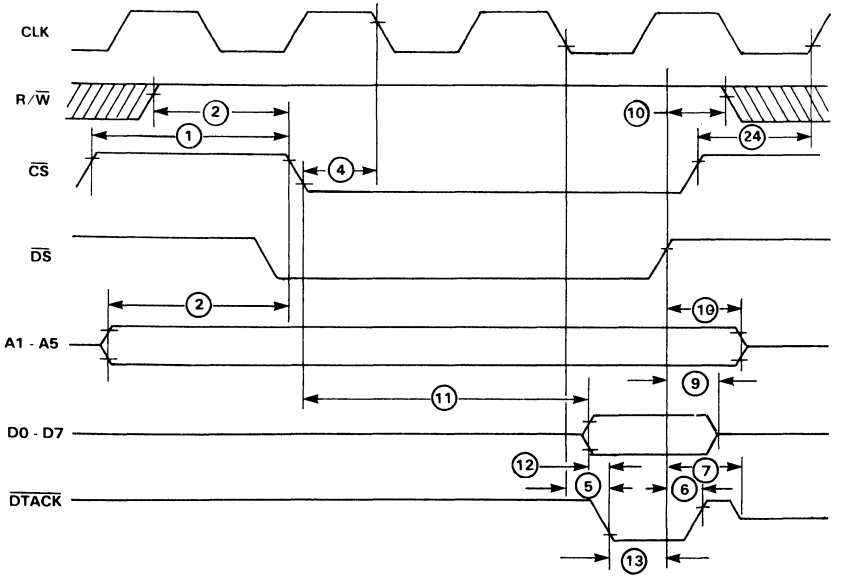
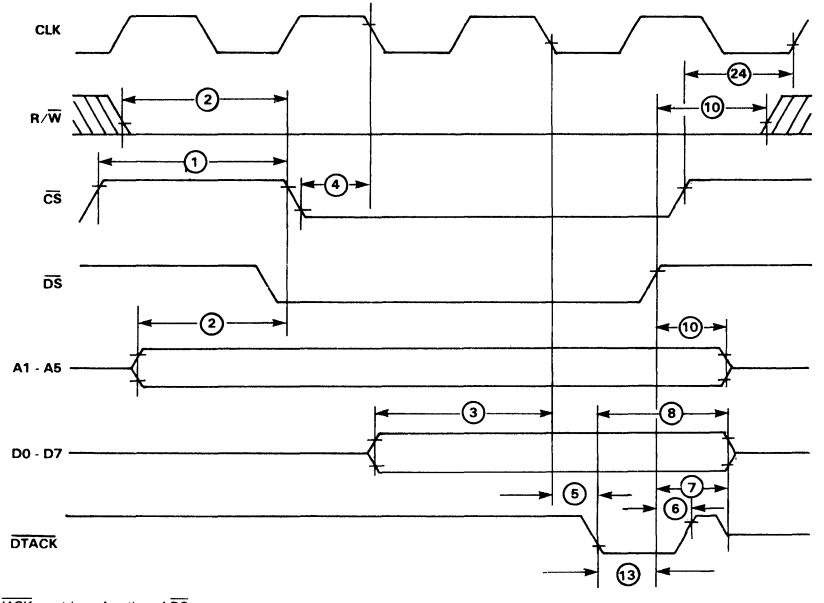


Figure 21. Read Cycle



NOTE:
$\overline{CS}$ and $\overline{IACK}$ must be a function of $\overline{DS}$.

Figure 22. Write Cycle

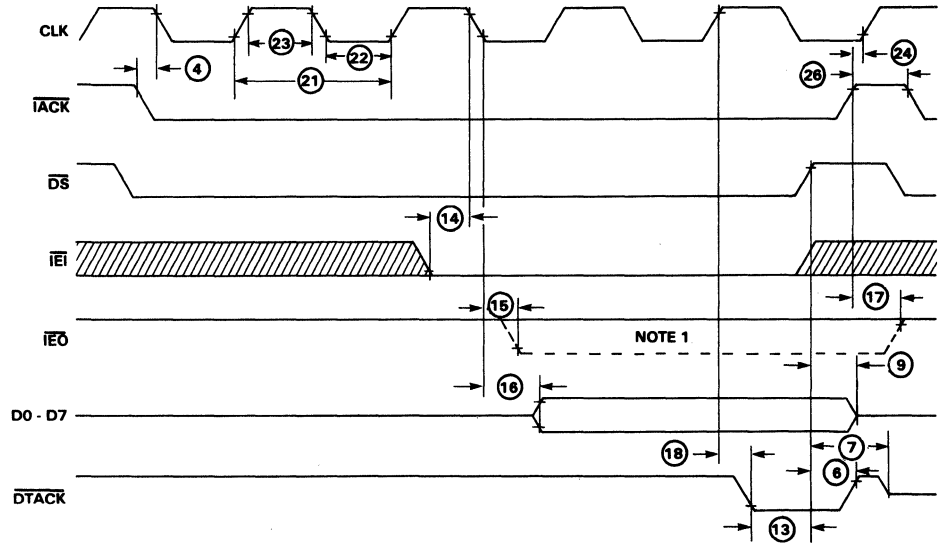
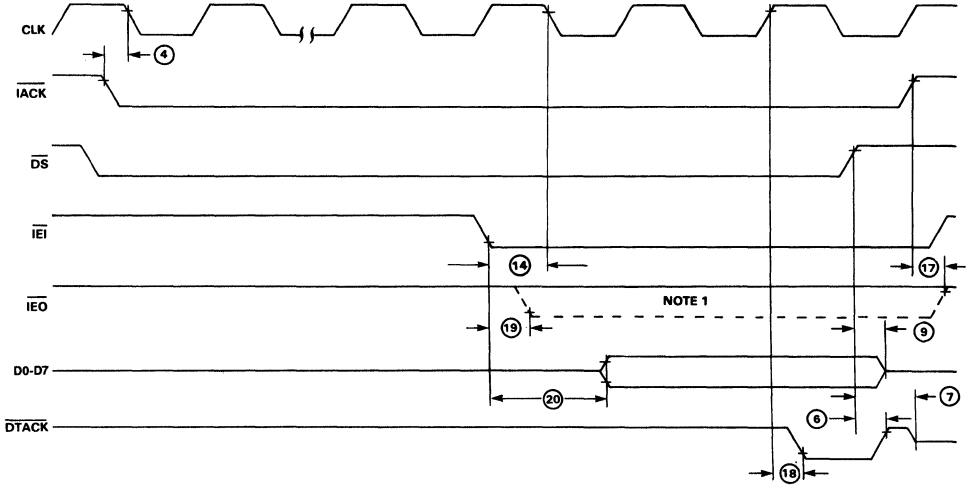
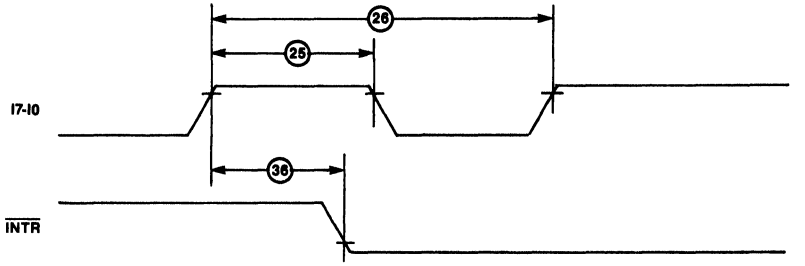


Figure 23. Interrupt Acknowledge ($\overline{IEI}$ Low)



NOTE:
$\overline{CS}$ and $\overline{IACK}$ must be a function of $\overline{DS}$.

Figure 24. Interrupt Acknowledge Cycle ($\overline{IEI}$ High)



NOTE: Active edge is assumed to be the rising edge.

Figure 25. Interrupt Timing

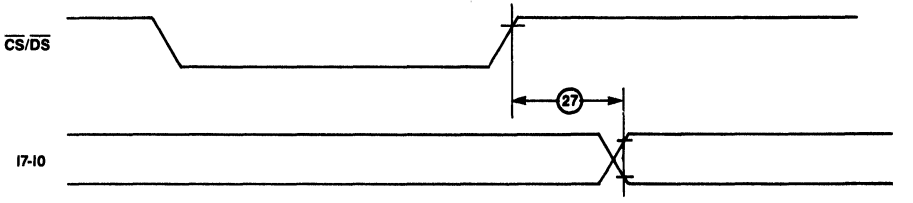


Figure 26. Port Timing

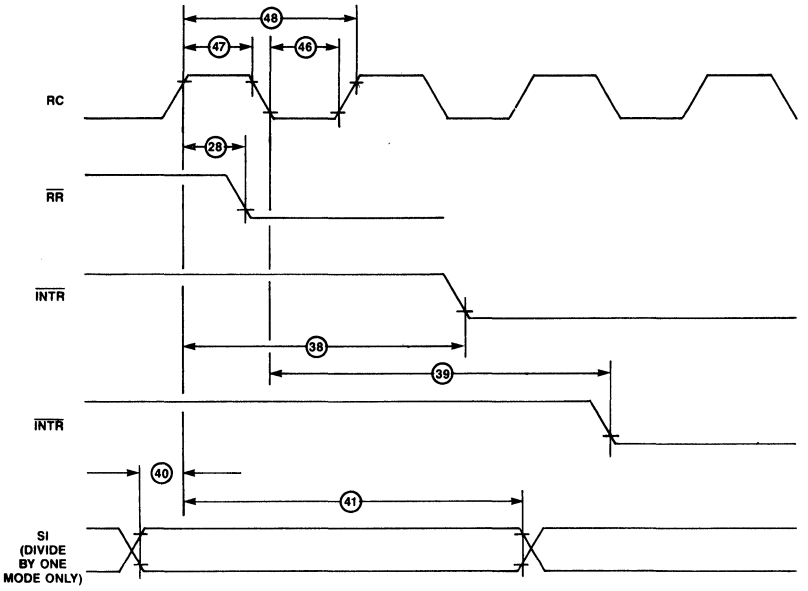


Figure 27. Receiver Timing

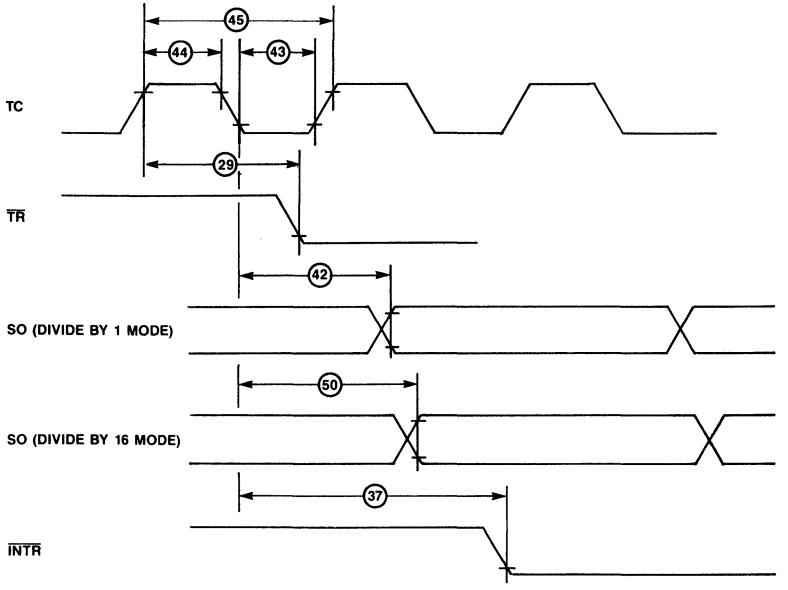


Figure 28. Transmitter Timing

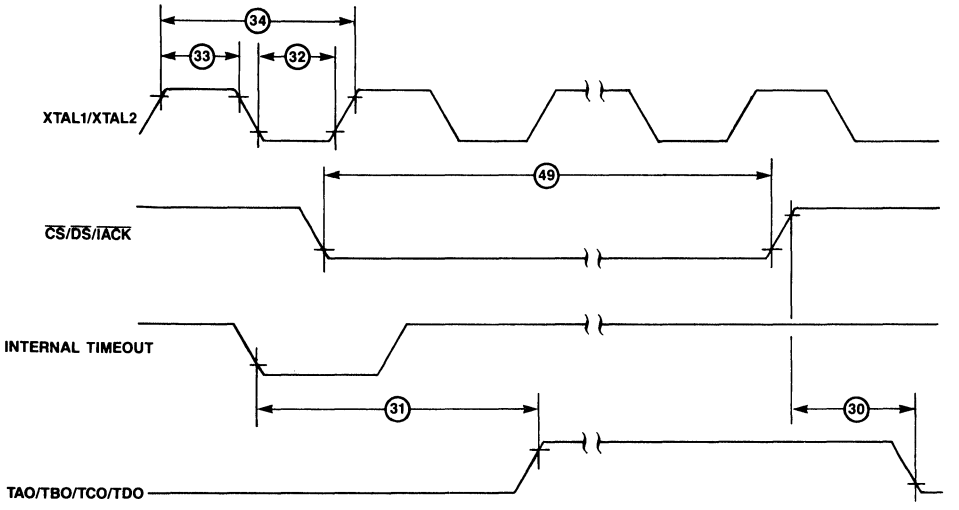


Figure 29. Timer Timing

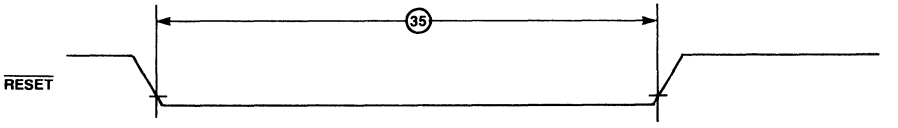
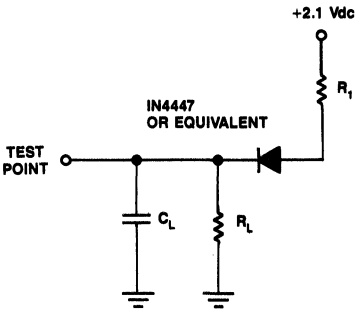


Figure 30. Reset Timing



for all outputs except $\overline{DTACK}$

C_L = 100 pf
R_L = 20 k Ω
R_1 = 180 Ω

for $\overline{DTACK}$

C_L = 130 pf
R_L = 6 k Ω
R_1 = 470 Ω

Figure 31. Typical Output

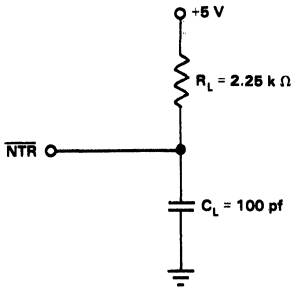
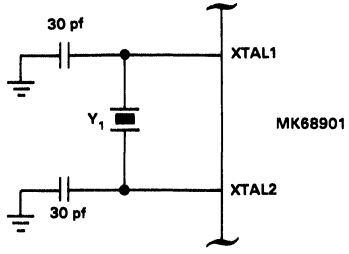


Figure 32. $\overline{INTR}$ Test Load



CRYSTAL PARAMETERS:
Parallel resonance, fundamental mode AT cut
$R_S \leq 150\ \Omega$ (F_R = 2.8 - 5.0 MHz);
$R_S \leq 300\ \Omega$ (F_R = 2.0 - 2.7 MHz)
C_L = 18 pf; C_M = 0.02 pf; C_h = 5 pf; L_M = 96 mH
F_R (typ) = 2.4576 MHz

Figure 33. MK68901 MFP External Oscillator Components

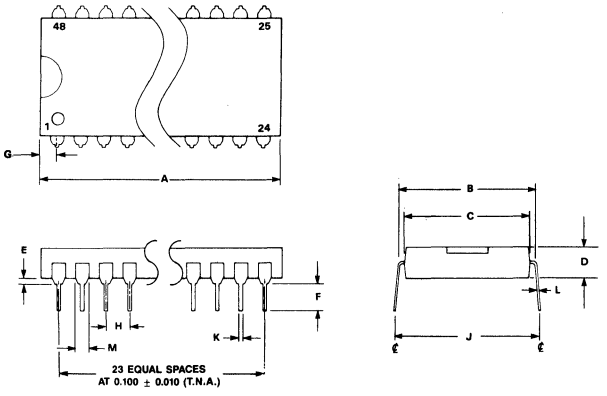
MK68901 ORDERING INFORMATION

PART NO.	PACKAGE TYPE	MAX. CLOCK FREQUENCY	TEMPERATURE RANGE
68901P00	Ceramic DIP	4.0 MHz	0° to 70°C
68901P05	Ceramic DIP	5.0 MHz	0° to 70°C
68901N00	Plastic DIP	4.0 MHz	0° to 70°C
68901N05	Plastic DIP	5.0 MHz	0° to 70°C
68901Q00[1]	Plastic PLCC	4.0 MHz	0° to 70°C
68901Q05[1]	Plastic PLCC	5.0 MHz	0° to 70°C

NOTE: 1. Contact Mostek for availability

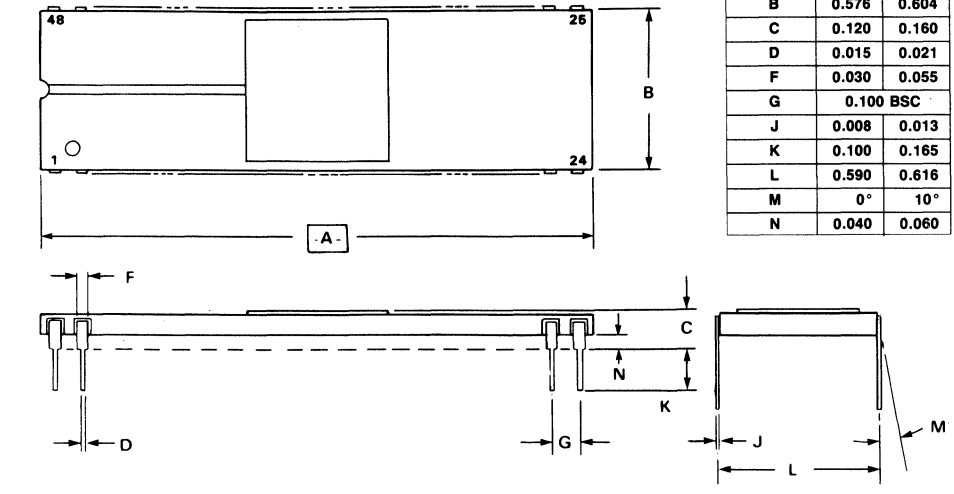
MK68901 48-Pin Plastic Dual-In-Line Package (N)

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	61.468	62.738	2.420	2.470
B	14.986	16.256	.590	.640
C	13.462	13.97	.530	.550
D	3.556	4064	.140	.160
E	0.381	1.524	.015	060
F	3048	3.81	.120	.150
G	1.524	2.286	.060	.090
H	1.186	1.794	.090	.110
J	15.24	17.78	.600	.700
K	0.381	0.533	.015	.021
L	0.203	0.305	.008	.012
M	1.143	1.778	.045	.070

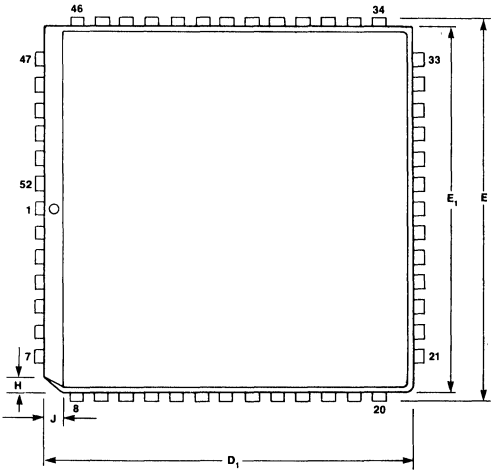


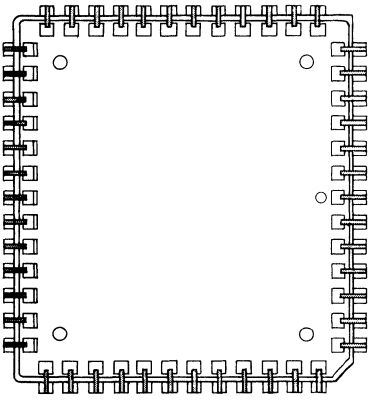
MK68901 48-Pin Ceramic Dual-In-Line Package (P)

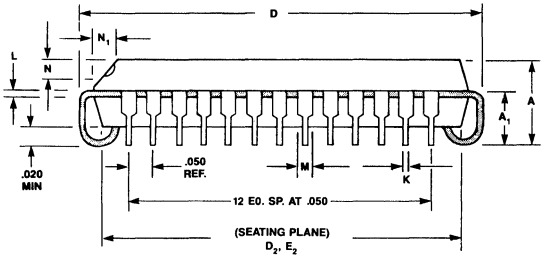
DIM.	INCHES	
	MIN	MAX
A	2.376	2.424
B	0.576	0.604
C	0.120	0.160
D	0.015	0.021
F	0.030	0.055
G	0.100 BSC	
J	0.008	0.013
K	0.100	0.165
L	0.590	0.616
M	0°	10°
N	0.040	0.060



MK68901 52-Pin
Plastic Leaded Chip Carrier (Q)







DIM.	INCHES	
	MIN	MAX
A	.165	.180
A_1	.090	.130
D	.785	.795
D_1	.750	.756
D_2	.690	.730
E	.785	.795
E_1	.750	.756
E_2	.690	.730
H	.042	.048
J	.042	.048
K	.013	.024
L	.008	.014
M	.026	.032
N/N_1	.043	.048

MK68901 PIN CONNECTIONS

PLCC	DIP	FUNC.	PLCC	DIP	FUNC.	PLCC	DIP	FUNC.
1	—	NC	19	17	XTAL1	37	33	$\overline{IEO}$
2	1	R/$\overline{W}$	20	18	XTAL2	38	34	$\overline{IEI}$
3	2	A1	21	—	NC	39	35	CLK
4	3	A2	22	19	TAI	40	36	GND
5	4	A3	23	20	TBI	41	37	D0
6	5	A4	24	21	$\overline{RESET}$	42	38	D1
7	6	A5	25	22	I0	43	39	D2
8	7	TC	26	23	I1	44	40	D3
9	8	SO	27	24	I2	45	41	D4
10	9	SI	28	25	I3	46	42	D5
11	10	RC	29	26	I4	47	43	D6
12	11	V_{CC}	30	27	I5	48	44	D7
13	—	NC	31	28	I6	49	45	$\overline{IACK}$
14	12	NC	32	29	I7	50	46	$\overline{DTACK}$
15	13	TAO	33	—	NC	51	47	$\overline{DS}$
16	14	TBO	34	30	$\overline{TR}$	52	48	$\overline{CS}$
17	15	TCO	35	31	$\overline{RR}$			
18	16	TDO	36	32	$\overline{INTR}$			

NOTE: NC - No Connection

THOMSON COMPONENTS MOSTEK

TS68HC901

HCMOS MULTI-FUNCTION PERIPHERAL

ADVANCE INFORMATION

The TS68HC901 muli-function peripheral (CMFP) is a member of the 68000 Family of peripherals and the CMOS version of the MK68901. The CMFP directly interfaces to the 68000 processor via an asynchronous bus structure and can also support both multiplexed and non multiplexed buses. Both vectored, non vectored and polled interrupt schemes are supported, with the CMFP providing unique vector number generation for each of its 16 interrupt sources. Additionally, handshake lines are provided to facilitate DMAC interfacing.

The TS68HC901 performs many of the functions common to most microprocessor-based systems. The resources available to the user include :

- Eight Individually Programmable I/O Pins with Interrupt Capability
- 16-Source Interrupt Controller with Individual Source Enabling and Masking
- Four Timers, Two of which are Multi-Mode Timers
- Timers may be used as Baud Rate Generators for the Serial Channel
- Single-Channel Full-Duplex Universal Synchronous/Asynchronous Receiver-Transmitter (USART) that Supports Asynchronous and with the Addition of a Polynominal Generator Checker Supports Byte Synchronous Formats.

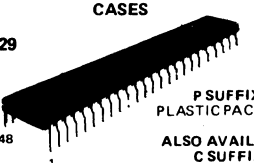
By incorporating multiple functions within the CMFP, the system designer retains flexibility while minimizing device count.
The CMOS technology used for the TS68HC901 reduces also the power consumption of the system.

HCMOS

MULTI-FUNCTION PERIPHERAL

CASES

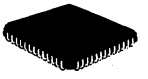
CB-229



P SUFFIX
PLASTIC PACKAGE

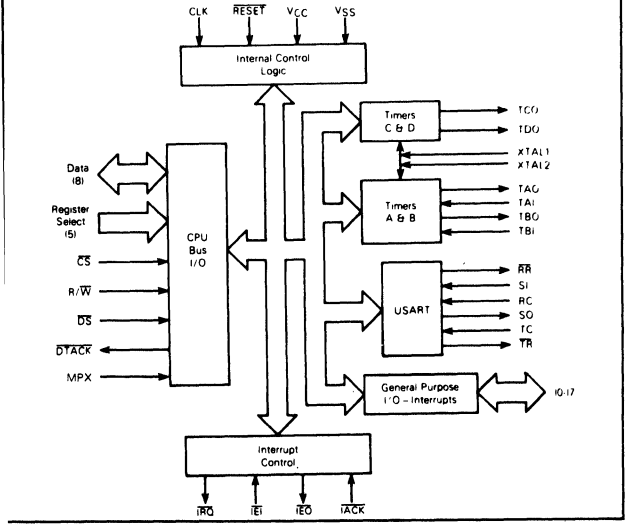
ALSO AVAILABLE
C SUFFIX
CERAMIC PACKAGE

CB-522

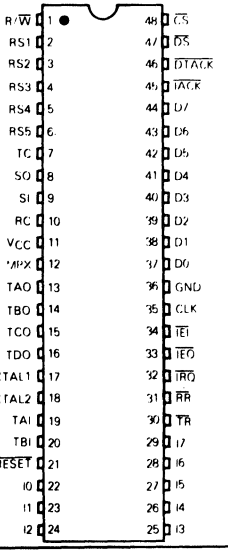


FN SUFFIX
PLCC 52

BLOCK DIAGRAM



PIN ASSIGNMENT



SECTION 1
INTRODUCTION

The TS68HC901 multi-function peripheral (CMFP) is a member of the 68000 peripherals. The CMFP directly interfaces to the 68000 processor via an asynchronous bus structure. Both vectored and polled interrupt schemes are supported, with the CMFP providing unique vector number generation for each of its 16 interrupt sources. Additionally, handshake lines are provided to facilitate DMAC interfacing. Refer to block diagram of the TS68HC901.

The TS68HC901 performs many of the functions common to most microprocessor-based systems.

The resources available to the user include:

- Eight Individually Programmable I/O Pins with Interrupt Capability
- 16-Source Interrupt Controller with Individual Source Enabling and Masking
- Four Timers, Two of which are Multi-Mode Timers
- Timers May Be Used as Baud Rate Generators for the Serial Channel
- Single-Channel Full-Duplex Universal Synchronous/Asynchronous Receiver-Transmitter (USART) that Supports Asynchronous and with the Addition of a Polynomial Generator Checker Supports Byte Synchronous Formats

By incorporating multiple functions within the CMFP, the system designer retains flexibility while minimizing device count.

rom a programmer's point of view, the versatility of the CMFP may be attributed to its register et. The registers are well organized and allow the CMFP to be easily tailored to a variety of pplications. All of the 24 registers are also directly addressable which simplifies programming. he register map is shown in Table 1-1.

Table 1-1. CMFP Register Map

Address							
	Binary						
Hex	RS5	RS4	RS3	RS2	RS1	Abbreviation	Register Name
01	0	0	0	0	0	GPIP	General Purpose I/O Register
03	0	0	0	0	1	AER	Active Edge Register
05	0	0	0	1	0	DDR	Data Direction Register
07	0	0	0	1	1	IERA	Interrupt Enable Register A
09	0	0	1	0	0	IERB	Interrupt Enable Register B
0B	0	0	1	0	1	IPRA	Interrupt Pending Register A
0D	0	0	1	1	0	IPRB	Interrupt Pending Register B
0F	0	0	1	1	1	ISRA	Interrupt In-Service Register A
11	0	1	0	0	0	ISRB	Interrupt In-Service Register B
13	0	1	0	0	1	IMRA	Interrupt Mask Register A
15	0	1	0	1	0	IMRB	Interrupt Mask Register B
17	0	1	0	1	1	VR	Vector Register
19	0	1	1	0	0	TACR	Timer A Control Register
1B	0	1	1	0	1	TBCR	Timer B Control Register
1D	0	1	1	1	0	TCDCR	Timers C and D Control Register
1F	0	1	1	1	1	TADR	Timer A Data Register
21	1	0	0	0	0	TBDR	Timer B Data Register
23	1	0	0	0	1	TCDR	Timer C Data Register
25	1	0	0	1	0	TDDR	Timer D Data Register
27	1	0	0	1	1	SCR	Synchronous Character Register
29	1	0	1	0	0	UCR	USART Control Register
2B	1	0	1	0	1	RSR	Receiver Status Register
2D	1	0	1	1	0	TSR	Transmitter Status Register
2F	1	0	1	1	1	UDR	USART Data Register

NOTE : Hex addresses assume that RS1 connects with A1, RS2 connects with A2, etc... and that DS is connected to LDS on the 68000 or DS is connected to DS on the 68008.

SECTION 2
SIGNAL AND BUS OPERATION DESCRIPTION

This section contains a brief description of the input and output signals. A discussion of bus operation during the various operations is also presented.

Note: The terms **assertion** and **negation** will be used extensively. This is done to avoid confusion when dealing with a mixture of "active low" and "active high" signals. The term assert or assertion is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. The term negate or negation is used to indicate that a signal is inactive or false.

2.1 SIGNAL DESCRIPTION

The input and output signals can be functionally organized into the groups shown in Figure 2-1. The following paragraphs provide a brief description of the signal and a reference (if applicable) to other sections that contain more detail about its function.

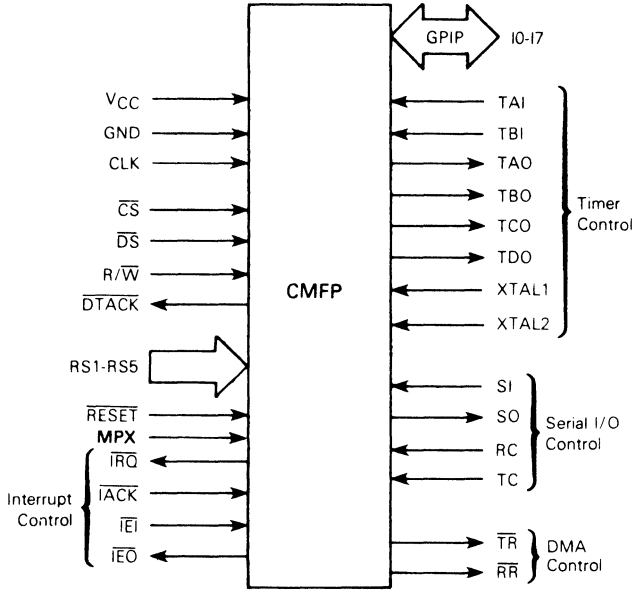


Figure 2-1. Input and Output Signals

2.1.1 V_{CC} and GND

These inputs supply power to the CMFP. The V_{CC} is power at + 5 volts and GND is the ground connection.

2.1.2 Clock (CLK)

The clock input is a single-phase TTL-compatible signal used for internal timing. This input should not be gated off at any time and must conform to minimum and maximum pulse width times. The clock is not necessarily the system clock in frequency nor phase. When the bus is multiplexed (MPX=1), an adress strobe signal is connected to this pin. In the non multiplexed mode (MPX=0), this input is connected to the system clock when used with a 68000 processor type or to V_{SS} (0 V_{DC}) when used with a 6800 processor type.

2.1.3 Asynchronous Bus Control

Asynchronous data transfers are controlled by chip select, data strobe, read/write, and data transfer acknowledge. The low order register select lines, RS1-RS5, select an internal CMFP register for a read or write operation. The reset line initializes the CMFP registers and the internal control signals.

2.1.3.1 CHIP SELECT ($\overline{CS}$). This input activates the CMFP for internal register access.

2.1.3.2 DATA STROBE ($\overline{DS}$). This input is part of the internal chip select and interrupt acknowl edge functions. The CMFP must be located on the lower portion of the 16-bit data bus so that th vector number passed to the processor during an interrupt acknowledge cycle will be located i the low byte of the data word. As a result, $\overline{DS}$ must be connected to the processor's lower dat strobe if vectored interrupts are to be used. Note that this forces all registers to be located at od addresses and latches data on the rising edge for writes. This signal is used as $\overline{RD}$ with an Inte processor type.

2.1.3.3 READ/WRITE (R/$\overline{W}$). This input defines a data transfer as a read (high) or a write (low cycle. This signal is used as $\overline{WR}$ with an Intel processor type.

2.1.3.4. DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$). This output signals the completio of the operation phase of a bus cycle to the processor. If the bus cycle is a processor read, th CMFP asserts DTACK to indicate that the information on the data bus is valid. If the bus cycle is processor to the CMFP, $\overline{DTACK}$ acknowledges the acceptance of the data by the CMFP. $\overline{DTACK}$ will be asserted only by an CMFP that has $\overline{CS}$ or $\overline{IACK}$ (and $\overline{IEI}$) asserted. This signal is not use with a 6800 processor type.

2.1.3.5 REGISTER SELECT BUS (RS1 THROUGH RS5). The lower five bits of the register sele bus select an internal CMFP register during a read or write operation.

.1.3.6 DATA BUS (D0 THROUGH D7). This bidirectional bus is used to receive data from or ransmit data to the CMFP's internal registers during a processor read or write cycle. During an nterrupt acknowledge cycle, the data bus is used to pass a vector number to the processor. Since the CMFP is an 8-bit peripheral, the CMFP could be located on either the upper or lower portion of the 16-bit data bus (even or odd address). However, during an interrupt acknowledge cycle, the vector number passed to the processor must be located in the low byte of the data word. As a result, D0-D7 of the CMFP must be connected to the low order eight bits of the processor data bus, placing CMFP registers at odd addresses if vectored interrupts are to be used.

2.1.3.7 RESET ($\overline{\text{RESET}}$). This input will initialize the CMFP during power up or in response to a total system reset. Refer to **2.2.3** for further information.

2.1.3.8 MPX. This signal selects the data bus mode:
MPX=0: non multiplexed mode
MPX=1: multiplexed mode. The register select lines RS1-RS5 and the data bus D0-D7 are multiplexed. An adress strobe must be connected to the CLK pin.

.1.4 Interrupt Control

The interrupt request and interrupt acknowledge signals are handshake lines for a vectored interrupt scheme. Interrupt enable in and the interrupt enable out implement a daisy-chained interrupt structure.

.1.4.1 INTERRUPT REQUEST ($\overline{\text{IRQ}}$). This output signals the processor that an interrupt is pending from the CMFP. These are 16 interrupt channels that can generate an interrupt request. Clearing the interrupt pending registers (IPRA and IPRB) or clearing the interrupt mask registers (IMRA and IMRB) will cause $\overline{\text{IRQ}}$ to be negated. $\overline{\text{IRQ}}$ will also be negated as the result of an interrupt acknowledge cycle, unless additional interrupts are pending in the CMFP. Refer to **SECTION 3** for further information.

2.1.4.2 INTERRUPT ACKNOWLEDGE ($\overline{\text{IACK}}$). If both $\overline{\text{IRQ}}$ and $\overline{\text{IEI}}$ are active, the CMFP will begin an interrupt acknowledge cycle when $\overline{\text{IACK}}$ and $\overline{\text{DS}}$ are asserted. The CMFP will supply a unique vector number to the processor which corresponds to the interrupt handler for the particular channel requiring interrupt service. In a daisy-chained interrupt structure, all devices in the chain must have a common $\overline{\text{IACK}}$. Refer to **2.2.2** and **3.1.2** for additional information.

.1.4.3 INTERRUPT ENABLE IN ($\overline{\text{IEI}}$). This input, together with the $\overline{\text{IEO}}$ signal, provides a daisy-chained interrupt structure for a vectored interrupt scheme. $\overline{\text{IEI}}$ indicates that no higher priority device is requesting interrupt service. So, the highest priority device in the chain should have its $\overline{\text{IEI}}$ pin tied low. During an interrupt acknowledge cycle, an CMFP with a pending interrupt is not allowed to pass a vector number to the processor until its $\overline{\text{IEI}}$ pin is asserted. When the daisy-chain option is not implemented, all CMFPs should have their $\overline{\text{IEI}}$ pin tied low. Refer to **3.2** for additional information.

2.1.4.4 INTERRUPT ENABLE OUT ($\overline{IEO}$). This output, together with the $\overline{IEI}$ signal, provides a daisy-chained interrupt structure for a vectored interrupt scheme. The $\overline{IEO}$ of a particular CMFP signals lower priority devices that neither the CMFP nor any other higher-priority device is requesting interrupt service. When a daisy-chain is implemented, $\overline{IEO}$ is tied to the next lower priority device's $\overline{IEI}$ input. The lowest priority device's $\overline{IEO}$ is not connected. When the daisy-chain option is not implemented, $\overline{IEO}$ is not connected. Refer to **3.2** for additional information.

2.1.5 General Purpose I/O Interrupt Lines (I0 Through I7)

This is an 8-bit pin-programmable I/O port with interrupt capability. The data direction register (DDR) individually defines each line as either a high-impedance input or a TTL-compatible output. As an input, each line can generate an interrupt on the user selected transition of the input signal. Refer to **SECTION 4** for further information.

2.1.6 Timer Control

These lines provide internal timing and auxiliary timer control inputs required for certain operating modes. Additionally, the timer outputs are included in this group.

2.1.6.1 TIMER CLOCK (XTAL1 AND XTAL2). This input provides the timing signal for the four timers. A crystal can be connected between the timer clock inputs, XTAL1 and XTAL2, or XTAL can be driven with a TTL-level clock while XTAL2 is not connected. The following crystal parameters are suggested:

a) Parallel resonance, fundamental mode AT-cut
b) Frequency tolerance measured with 18 picofarads load (0.1% accuracy) — drive level 1 microwatts
c) Shunt capacitance equals 7 picofarads maximum
d) Series resistance:
   $2.0 < f < 2.7$ MHz; $R_S \leq 300\ \Omega$
   $2.8 < f < 4.0$ MHz; $R_S \leq 150\ \Omega$

2.1.6.2 TIMER INPUTS (TAI AND TBI). These inputs are control signals for timers A and B in the pulse width measurement mode and event count mode. These signals generate interrupts at the same priority level as the general purpose I/O interrupt lines I4 and I3, respectively. While I4 and I3 do not have interrupt capability when the timers are operated in the pulse width measurement mode or the event count mode, I4 and I3 may still be used for I/O. Refer to **5.1.2** and **5.1.3** for further information.

2.1.6.3 TIMER OUTPUTS (TAO, TBO, TCO, AND TDO). Each timer has an associated output which toggles when its main counter counts through 01 (hexadecimal), regardless of which operational mode is selected. When in the delay mode, the timer output will be a square wave with a period equal to two timer cycles. This output signal may be used to supply the universal synchronous/asynchronous receiver-transmitter (USART) baud rate clocks. Timer outputs TAO and TBO may be cleared at any time by writing a one to the reset location in timer control registers A and B. Also, a device reset forces all timer outputs low. Refer to **5.2.2** for additional information.

2.1.7 Serial I/O Control

The full duplex serial channel is implemented by a serial input and output line. The independent receive and transmit sections may be clocked by separate timing signals on the receiver clock input and the transmitter clock input.

2.1.7.1 SERIAL INPUT (SI). This input line is the USART receiver data input. This input is not used in the USART loopback mode. Refer to **6.3.2** for additional information.

2.1.7.2 SERIAL OUTPUT (SO). This output line is the USART transmitter data output. This output is driven high during a device reset.

2.1.7.3 RECEIVER CLOCK (RC). This input controls the serial bit rate of the receiver. This signal may be supplied by the timer output lines or by any external TTL-level clock which meets the minimum and maximum cycle times. This clock is not used in the USART loopback mode. Refer to 6.3.2 for additional information.

2.1.7.4 TRANSMITTER CLOCK (TC). This input controls the serial bit rate of the transmitter. This signal may be supplied by the timer output lines or by an external TTL-level clock which meets the minimum and maximum cycle times.

2.1.8 DMA Control

The USART supports DMA transfers through its receiver ready and transmitter ready status lines.

2.1.8.1 RECEIVER READY ($\overline{RR}$). This output reflects the receiver buffer full status for DMA operations.

2.1.8.2 TRANSMITTER READY ($\overline{TR}$). This output reflects the transmitter buffer empty status for DMA operations.

2.1.9 Signal Summary

Table 2-1 is a summary of all the signals discussed in the previous paragraphs.

Table 2-1. Signal Summary

Signal Name	Mnemonic	I/O	Active
Power Input	V_{CC}	Input	High
Ground	GND	Input	Low
Clock	CLK	Input	N/A
Chip Select	$\overline{CS}$	Input	Low
Data strobe	$\overline{DS}$	Input	Low
Read/Write	R/$\overline{W}$	Input	Read – High, Write – Low
Data Transfer Acknowledge	$\overline{DTACK}$	Output	Low
Register Select Bus	RS1-RS5	Input	N/A
Data Bus	D0-D7	I/O	N/A
Reset	$\overline{RESET}$	Input	Low
Interrupt Request	$\overline{IRQ}$	Output	Low
Interrupt Acknowledge	$\overline{IACK}$	Input	Low
Interrupt Enable In	$\overline{IEI}$	Input	Low
Interrupt Enable Out	$\overline{IEO}$	Output	Low
General Purpose I/O – Interrupt Lines	I0-I7	I/O	N/A
Timer Clock	XTAL1, XTAL2	Input	High
Timer Inputs	TAI, TBI	Input	N/A
Timer Outputs	TAO,TBO, TCO, TD0	Output	N/A
Serial Input	SI	Input	N/A
Serial Output	SO	Output	N/A
Receiver Clock	RC	Input	N/A
Transmitter Clock	TC	Input	N/A
Receiver Ready	$\overline{RR}$	Output	Low
Transmitter Ready	$\overline{TR}$	Output	Low
MPX	MPX	Input	N/A

2.2 BUS OPERATION

The following paragraphs explain the control signals and bus operation during data transfer opera tions and reset.

2.2.1 Data Transfer Operations

Transfer of data between devices involves the following pins:

Register Select Bus – RS1 through RS5

Data Bus – D0 through D7

Control Signals

The address and data buses are separate parallel buses used to transfer data using an asynchronou bus structure. In all cycles, the bus master assumes responsibility for deskewing all signals it issue at both the start and end of a cycle. Additionally, the bus master is responsible for deskewing th acknowledge and data signals from the peripheral devices.

2.2.1.1 **READ CYCLE.** To read an CMFP register, $\overline{CS}$ and $\overline{DS}$ must be asserted, and R/$\overline{W}$ must be high. The CMFP will place the contents of the register which is selected by the register select bus (RS1 through RS5) on the data bus (D0 through D7) and then assert $\overline{DTACK}$. The register addresses are shown in Table 1-1.

After the processor has latched the data, $\overline{DS}$ is negated. The negation of either $\overline{CS}$ or $\overline{DS}$ will terminate the read operation. The CMFP will drive $\overline{DTACK}$ high and place it in the high-impedance state. Also, the data bus will be in the high-impedance state. The timing for a read cycle is shown in Figure 2-2. Refer to **7.7** for actual timing numbers.

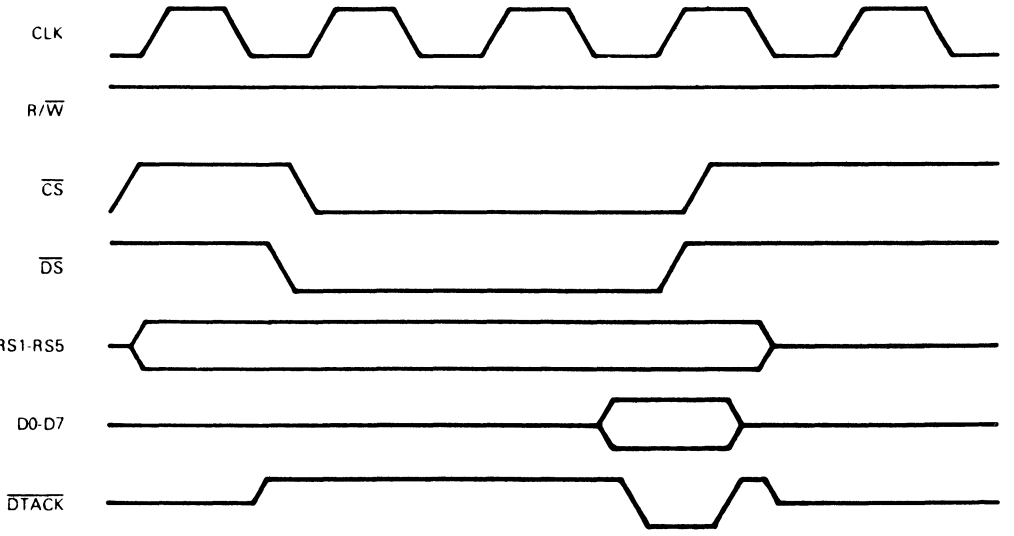


Figure 2-2. Read Cycle Timing

2.2.1.2 **WRITE CYCLE.** To write a register, $\overline{CS}$ and $\overline{DS}$ must be asserted, and R/$\overline{W}$ must be low. The CMFP will decode the address bus to determine which register is selected (the register map is shown in Table 1-1). Then the register will be loaded with the contents of the data bus and $\overline{DTACK}$ will be asserted.

When the processor recognizes $\overline{DTACK}$, $\overline{DS}$ will be negated. The write cycle is terminated when either $\overline{CS}$ or $\overline{DS}$ is negated. The CMFP will drive $\overline{DTACK}$ high and place it in the high-impedance state. The timing for a write cycle is shown in Figure 2-3. Refer to **7.7** for actual numbers.

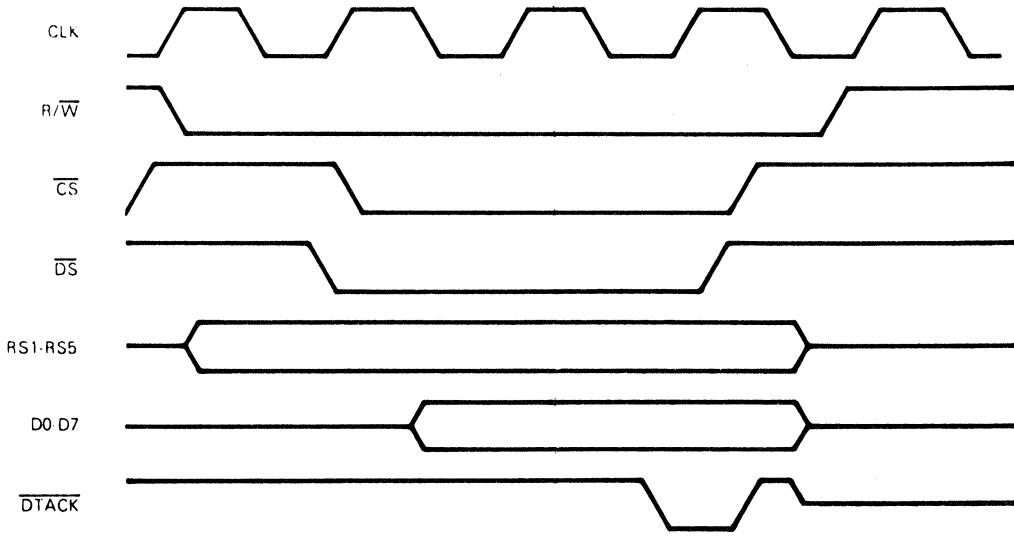


Figure 2-3. Write Cycle Timing

2.2.2 Interrupt Acknowledge Operation

The CMFP has 16 interrupt sources, eight internal sources, and eight external sources. When an interruptrequest is pending, the CMFP will assert $\overline{IRQ}$. In a vectored interrupt scheme, the processor will acknowledge the interrupt request by performing an interrupt acknowledge cycle. $\overline{IACK}$ and $\overline{DS}$ will be asserted. The CMFP responds to the $\overline{IACK}$ signal by placing a vector number on the lower eight bits of the data bus. This vector number corresponds to the $\overline{IRQ}$ handler for the particular interrupt requesting service. The format of this vector number is given in Figure 3-1.

When the CMFP asserts $\overline{DTACK}$ to indicate that valid data is on the bus, the processor will latch the data and terminate the bus cycle by negating $\overline{DS}$. When either $\overline{DS}$ or $\overline{IACK}$ are negated, the CMFP will terminate the interrupt acknowledge operation by driving $\overline{DTACK}$ high and placing it in the high-impedance state. Also, the data bus will be placed in the high-impedance state. $\overline{IRQ}$ will be negated as a result of the $\overline{IACK}$ cycle unless additional interrupts are pending.

The CMFP can be part of a daisy-chain interrupt structure which allows multiple CMFPs to be placed at the same interrupt level by sharing a common $\overline{IACK}$ signal. A daisy-chain priority scheme is implemented with signals $\overline{IEI}$ and $\overline{IEO}$. $\overline{IEI}$ indicates that no higher priority device is requesting interrupt service. $\overline{IEO}$ signals lower priority devices that neither this device nor any higher priority device is requesting service. To daisy-chain CMFPs, the highest priority CMFP has its $\overline{IEI}$ tied low and successive CMFPs have their $\overline{IEI}$ connected to the next higher priority device's $\overline{IEO}$. Note that when the daisy-chain interrupt structure is not implemented, the $\overline{IEI}$ of all CMFPs must be tied low. Refer to **3.2** for additional information.

When the processor initiates an interrupt acknowledge cycle by driving $\overline{IACK}$ and $\overline{DS}$, the CMFP whose $\overline{IEI}$ is low may respond with a vector number if an interrupt is pending. If this device does not have a pending interrupt, $\overline{IEO}$ is asserted which allows the next lower priority device to respond to the interrupt acknowledge. When an CMFP propagates $\overline{IEO}$, it will not drive the data bus nor $\overline{DTACK}$ during the interrupt acknowledge cycle. The timing for an $\overline{IACK}$ cycle is shown in Figure 2-4. Refer to **7.6** for further information.

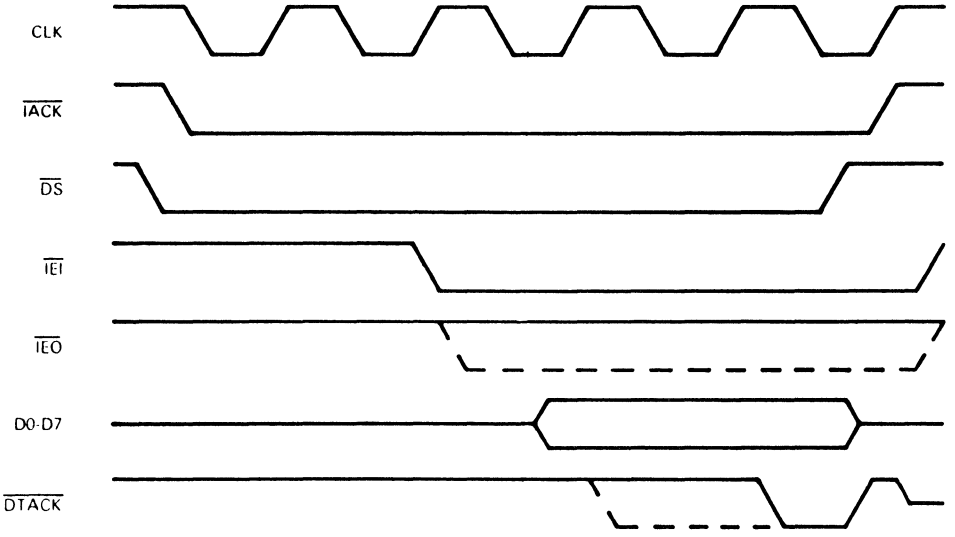


Figure 2-4. $\overline{IACK}$ Cycle Timing

2.2.3 Reset Operation

The reset operation will initialize the CMFP to a known state. The reset operation requires that the $\overline{RESET}$ input be asserted for a minimum of two microseconds. During a device reset condition, all internal CMFP registers are cleared except for the timer data registers (TADR, TBDR, TCDR, and TDDR), the USART data register (UDR), the transmitter status register (TSR) and the interrupt vector register. All timers are stopped and the USART receiver and transmitter are disabled. The interrupt channels are also disabled and any pending interrupts are cleared. In addition, the general purpose interrupt I/O lines are placed in the high-impedance input mode and the timer outputs are driven low. External CMFP signals are negated. The interrupt vector register is initialized to a $0F.

2.2.4 Non Multiplexed mode

In this mode the MPX input must be set to zero, and the TS68HC901 can be used with a 68000 processor type or a 6800 processor type. Refer to figure 7-4, 7-5, 7-8 for the electrical characteristics.

With a 6800 processor type the $\overline{DS}$ pin is connected to the E signal of the processor, the $\overline{DTACK}$ signal is not used and the CLK must be zeroed.

2.2.5 Multiplexed mode

The CMFP can be used either on a MOTOROLA or INTEL bus type. In this case the MPX pin connected to V_{CC}.The following table gives the signification of the different signals used. dummy access to the TS68HC901 has to be done before any valid access in order to set up th internal logic of sampling.

Pin	MOTOROLA 6800 type	MOTOROLA Multiplexed	INTEL
48	$\overline{CS}$	$\overline{CS}$	$\overline{CS}$
47	E	$\overline{DS}$	$\overline{RD}$
1	R/$\overline{W}$	R/$\overline{W}$	$\overline{WR}$
35	V_{SS}	AS	ALE

SECTION 3
INTERRUPT STRUCTURE

In a 68000 system, the CMFP will be assigned to one of the seven possible interrupt levels. All interrupt service requests from the CMFP's 16 interrupt channels will be presented at this level. Although, as an interrupt controller, the CMFP will internally prioritize its 16 interrupt sources. Additional interrupt sources may be placed at the same interrupt level by daisy-chaining multiple CMFPs. The CMFPs will be prioritized by their position in the chain.

3.1 INTERRUPT PROCESSING

Each CMFP provides individual interrupt capability for its various functions. When an interrupt is received on one of the external interrupt channels or from one of the eight internal sources, the CMFP will request interrupt service. The 16 interrupt channels are assigned a fixed priority so that multiple pending interrupts are serviced according to their relative importance. Since the CMFP can internally generate 16 vector numbers, the unique vector number which corresponds to the highest priority channel that has a pending interrupt is presented to the processor during an interrupt acknowledge cycle. This unique vector number allows the processor to immediately begin execution of the interrupt handler for the interrupting source, decreasing interrupt latency time.

3.1.1 Interrupt Channel Prioritization

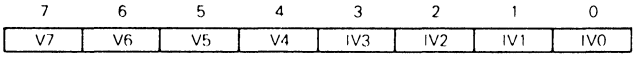
The 16 interrupt channels are prioritized as shown in Table 3-1. General purpose interrupt 7 (I7) is the highest priority interrupt channel and I0 is the lowest priority channel. Pending interrupts are presented to the CPU in order of priority unless they have been masked off. By selectively masking interrupts, the channels are in effect re-prioritized.

Table 3-1. Interrupt Channel Prioritization

Priority	Channel	Description
Highest	1111	General Purpose Interrupt 7 (I7)
	1110	General Purpose Interrupt 6 (I6)
	1101	Timer A
	1100	Receiver Buffer Full
	1011	Receive Error
	1010	Transmit Buffer Empty
	1001	Transmit Error
	1000	Timer B
	0111	General Purpose Interrupt 5 (I5)
	0110	General Purpose Interrupt 4 (I4)
	0101	Timer C
	0100	Timer D
	0011	General Purpose Interrupt 3 (I3)
	0010	General Purpose Interrupt 2 (I2)
	0001	General Purpose Interrupt 1 (I1)
Lowest	0000	General Purpose Interrupt 0 (I0)

3.1.2 Interrupt Vector Number Format

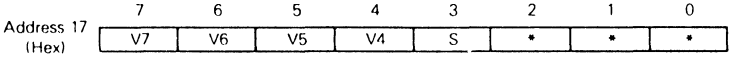
During an interrupt acknowledge cycle, a unique 8-bit vector number is presented to the system which corresponds to the specific interrupt source which is requesting service. The format of the vector is shown in Figure 3-1. The most significant four bits of the interrupt vector number are user programmable. These bits are set by writing the upper four bits of the vector register which is shown in Figure 3-2. The low order bits are generated internally by the TS68HC901. Note that the binary channel number shown in Table 3-1 corresponds to the low order bits of the vector number associated with each channel.



V7-V4 The four most significant bits are copied from the vector register

IV3-IV0 These bits are supplied by the CMFP. They are the binary channel number of the highest priority channel that is requesting interrupt service.

Figure 3-1. Interrupt Vector Format



*Unused bits are read as zero.

V7-V4 The upper four bits of the vector register are written by the user. These bits become the most significant four bits of the interrupt vector number.

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

S In-Service Register Enable. When the S bit is zero, the CMFP is in the automatic end-of-interrupt mode and the in-service register bits are forced low. When the S bit is a one, the CMFP is in the software end-of-interrupt mode and the in-Service register bits are enabled. Refer to **3.4.2** and **3.4.3** for additional information.
tion.

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

Figure 3-2 Vector Register Format (VR)

3.2 DAISY-CHAINING CMFPs

As an interrupt controller, the TS68HC901 CMFP will support eight external interrupt sources in addition to its eight internal interrupt sources. When a system requires more than eight external interrupt sources to be placed at the same interrupt level, sources may be added to the prioritized structure by daisy-chaining CMFPs. Interrupt sources are prioritized internally within each CMFP and the CMFPs are prioritized by their position in the chain. Unique vector numbers are provided for each interrupt source.

The $\overline{IEI}$ and $\overline{IEO}$ signals implement the daisy-chained interrupt structure. The $\overline{IEI}$ of the highest priority CMFP is tied low and the $\overline{IEO}$ output of this device is tied to the next highest priority CMFP's $\overline{IEI}$. The $\overline{IEI}$ and $\overline{IEO}$ signals are daisy-chained in this manner for all CMFPs in the chain, with the lowest priority CMFP's $\overline{IEO}$ left unconnected. A diagram of an interrupt daisy-chain is shown in Figure 3-3.

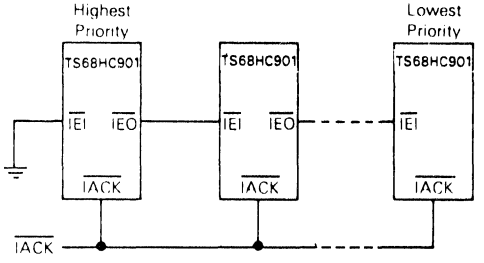


Figure 3-3. Daisy-Chained Interrupt Structure

Daisy-chaining requires that all parts in the chain have a common $\overline{IACK}$. When the common $\overline{IACK}$ is asserted during an interrupt acknowledge cycle, all parts will prioritize interrupts in parallel. When the $\overline{IEI}$ signal to a CMFP is asserted, the part may respond to the $\overline{IACK}$ cycle if it requires interrupt service. Otherwise, the part will assert $\overline{IEO}$ to the next lower priority device. Thus, priority is passed down the chain via $\overline{IEI}$ and $\overline{IEO}$ until a part which has a pending interrupt is reached. The part with the pending interrupt passes a vector number to the processor and does not propagate $\overline{IEO}$.

3.3 INTERRUPT CONTROL REGISTERS

CMFP interrupt processing is managed by the interrupt enable registers A and B, interrupt pending registers A and B, and interrupt mask registers A and B. These registers allow the programmer to enable or disable individual interrupt channels, mask individual interrupt channels, and access pending interrupt status information. In-service registers A and B allow interrupts to be nested as described in **3.4**. The interrupt control registers are shown in Figure 3-4.

3.3.1 Interrupt Enable Registers

The interrupt channels are individually enabled or disabled by writing a one or zero, respectively, to the appropriate bit of interrupt enable register A (IERA) or interrupt enable register B (IERB). The processor may read these registers at any time.

When a channel is enabled, interrupts received on the channel will be recognized by the CMFP and $\overline{IRQ}$ will be asserted to the processor, indicating that interrupt service is required. On the other hand, a disabled channel is completely inactive; interrupts received on the channel are ignored by the CMFP.

Writing a zero to a bit of interrupt enable register A or B will cause the corresponding bit of interrupt pending register A or B to be cleared. This will terminate all interrupt service requests for the channel and also negate $\overline{IRQ}$, unless interrupts are pending from other sources. Disabling a channel, however, does not affect the corresponding bit in interrupt in-service registers A or B. So, if the CMFP is in the software end-of-interrupt mode (see **3.4.3**) and an interrupt is in service when a channel is disabled, the in-service status bit for that channel will remain set until cleared by software.

(a) Interrupt Enable Registers (IERA and IERB)

	7	6	5	4	3	2	1	0
Address 07 (Hex)	GPIP7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

	7	6	5	4	3	2	1	0
Address 09 (Hex)	GPIP5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, the associated interrupt channel is disabled. When a bit is a one, the associated interrupt channel is enabled.

SET a) MPU writes a one

CLEARED a) MPU writes a zero
b) Reset

(b) Interrupt Pending Registers (IPRA and IPRB)

	7	6	5	4	3	2	1	0
Address 0B (Hex)	GPIP7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

	7	6	5	4	3	2	1	0
Address 0D (Hex)	GPIP5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, no interrupt is pending on the associated interrupt channel. When a bit is a one, an interrupt is pending on the associated interrupt channel.

SET a) Interrupt is received on an enabled interrupt channel

CLEARED a) Interrupt vector for the associated interrupt channel is passed during an $\overline{\text{IACK}}$ cycle
b) Associated interrupt channel is disabled
c) MPU writes a zero
d) Reset

(c) Interrupt In-Service Registers (ISRA and ISRB)

	7	6	5	4	3	2	1	0
Address 0F (Hex)	GPIP7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

	7	6	5	4	3	2	1	0
Address 11 (Hex)	GPIP5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, no interrupt processing is in progress for the associated interrupt channel. When a bit is a one, interrupt processing is in progress for the associated interrupt channel.

SET a) Interrupt vector number for the associated interrupt channel is passed during an $\overline{\text{IACK}}$ cycle and the S bit of the vector register is set.

CLEARED a) Interrupt service is completed for the associated interrupt channel
b) The S bit of the vector register is a zero.
c) MPU writes a zero
d) Reset

Figure 3-4. Interrupt Control Registers (Sheet 1 of 2)

(d) Interrupt Mask Registers (IMRA and IMRB)

	7	6	5	4	3	2	1	0
Address 13 (Hex)	GPIP7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

	7	6	5	4	3	2	1	0
Address 15 (Hex)	GPIP5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is zero, interrupts are masked for the associated interrupt channel. When a bit is a one, interrupts are not masked for the associated interrupt channel.

SET a) MPU writes a one

CLEARED a) MPU writes a zero
b) Reset

Figure 3-4. Interrupt Control Registers (Sheet 2 of 2)

3.3.2 Interrupt Pending Registers

When an interrupt is received on an enabled channel, the corresponding interrupt pending bit is set in interrupt pending register A or B (IPRA or IPRB). In a vectored interrupt scheme, this bit will be cleared when the processor acknowledges the interrupting channel and the CMFP responds with a vector number. In a polled interrupt system, the interrupt pending registers must be read to determine the interrupting channel and then the interrupt pending bit is cleared by the interrupt handling routine without performing an interrupt acknowledge sequence.

A single bit of the interrupt pending registers is cleared in software by writing ones to all bit positions except the bit to be cleared. Note that writing ones to IPRA and IPRB has no effect on the contents of the register. A single bit of the interrupt pending registers is also cleared when the corresponding channel is disabled by writing a zero to the appropriate bit of IERA or IERB.

3.3.3 Interrupt Mask Registers

Interrupts are masked for a channel by clearing the appropriate bit in interrupt mask register A or B (IMRA or IMRB). Even though an enabled channel is masked, the channel will recognize subsequent interrupts and set its interrupt pending bit. However, the channel is prevented from requesting interrupt service ($\overline{IRQ}$ to the processor) as long as the mask bit for that channel is cleared.

If a channel is requesting interrupt service at the time that its corresponding bit in IMRA or IMRB is cleared, the request will cease and $\overline{IRQ}$ will be negated, unless another channel is requesting interrupt service. Later, when the mask bit is set, any pending interrupt on the channel will be processed according to the channel's assigned priority. IMRA and IMRB may be read at any time.

3.4 NESTING CMFP INTERRUPTS

In a 68000 vectored interrupt system, the CMFP is assigned to one of seven possible interrupt levels. When an interrupt is received from the CMFP, an interrupt acknowledge for that level is initiated. Once an interrupt is recognized at a particular level, interrupts at that same level or

below are masked by 68000. As long as the processor's interrupt mask is unchanged, the 68000 interrupt structure will prohibit the nesting of interrupts at the same interrupt level. However, additional interrupt requests from the CMFP can be recognized before a previous channel's interrupt service routine is completed by lowering the processor's interrupt mask to the next lower interrupt level within the interrupt handler.

When nesting CMFP interrupts, it may be desirable to permit interrupts on any CMFP channel, regardless of its priority, to preempt or delay interrupt processing of an earlier channel's interrupt service request. Or, it may be desirable to only allow subsequent higher priority channel interrupt requests to supercede previously recognized lower priority interrupt requests. The CMFP interrupt structure provides this flexibility by offering two end-of-interrupt options for vectored interrupt schemes. Note that the end-of-interrupt modes are not active in a polled interrupt scheme.

3.4.1 Selecting The End-Of-Interrupt Mode

In a vectored interrupt scheme, the CMFP may be programmed to operate in either the automatic end-of-interrupt mode or the software end-of-interrupt mode. The mode is selected by writing the S bit of the vector register (see Figure 3-2). When the S bit is programmed to a one, the CMFP is placed in the software end-of-structure mode and when the S bit is a zero, all channels operate in the automatic end-of-interrupt mode.

3.4.2 Automatic End-Of-Interrupt

When an interrupt vector number is passed to the processor during an interrupt acknowledge cycle, the corresponding channel's interrupt pending bit is cleared. In the automatic end-of-interrupt mode, no further history of the interrupt remains in the CMFP. The in-service bits of the interrupt in-service registers (ISRA and ISRB) are forced low. Subsequent interrupts which are received on any CMFP channel will generate an interrupt request to the processor, even if the current interrupt's service routine has not been completed.

3.4.3 Software End-Of-Interrupt

In the software end-of-interrupt mode, the channel's associated interrupt pending bit is cleared and in addition, the channel's in-service bit of in-service register A or B is set when its vector number is passed to the processor during an $\overline{\text{IACK}}$ cycle. A higher priority channel may subsequently request interrupt service and be acknowledged, but as long as the channel's in-service bit is set, no lower priority channel may request interrupt service nor pass its vector during an interrupt acknowledge sequence.

While only higher priority channels may request interrupt service, any channel can receive an interrupt and set its interrupt pending bit. Even the channel whose in-service bit is set can receive a second interrupt. However, no interrupt service request is made until its in-service bit is cleared.

The in-service bit for a particular channel can be cleared by writing a zero to its corresponding bit in ISRA or ISRB and ones to all other bit positions. Since bits in the in-service registers can only be cleared in software and not set, writing ones to the registers does not alter their contents. ISRA and ISRB may be read at any time.

SECTION 4
GENERAL PURPOSE INPUT/OUTPUT INTERRUPT PORT

The general purpose interrupt input/output (I/O) port (GPIP) provides eight I/O lines (I0 through I7) that may be operated as either inputs or outputs under software control. In addition, these lines may optionally generate an interrupt on either a positive transition or a negative transition of the input signal. The flexibility of the GPIP allows it to be configured as an 8-bit I/O port or for bit I/O. Since interrupts are enabled on a bit-by-bit basis, a subset of the GPIP could be programmed as handshake lines or the port could be connected to as many as eight external interrupt sources, which would be prioritized by the CMFP interrupt controller for interrupt service.

4.1 6800 INTERRUPT CONTROLLER

The CMFP interrupt controller is particularly useful in a system which has many 6800-type devices. Typically, in a vectored 68000 system, 6800-type peripherals use the autovector which corresponds to their assigned interrupt level since they do not provide a vector number in response to an $\overline{IACK}$ cycle. The autovector interrupt handler must then poll all 6800-type devices at that interrupt level to determine which device is requesting service. However, by tying the $\overline{IRQ}$ output from a 6800-type device to the general purpose I/O interrupt port (GPIP) of a CMFP, a unique vector number will be provided to the processor during an interrupt acknowledge cycle. This interrupt structure will significantly reduce interrupt latency for 6800-type devices and other peripheral devices which do not support vector-by-device.

4.2 GPIP CONTROL REGISTERS

The GPIP is programmed via three control registers shown in Figure 4-1. These registers control the data direction, provide user access to the port, and specify the active edge for each bit of the GPIP which will produce an interrupt. These registers are described in detail in the following paragraphs.

4.2.1 GPIP Data Register

The general purpose I/O data register is used to input or output data to the port. When data is written to the GPIP data register, those pins which are defined as inputs will remain in the high-impedance state. Pins which are defined as outputs will assume the state (high or low) of their corresponding bit in the data register. When the GPIP is read, data will be passed directly from the bits of the data register for pins which are defined as outputs. Data from pins defined as inputs will come from the input buffers.

4.2.2 Active Edge Register

The active edge register (AER) allows each of the GPIP lines to produce an interrupt on either a one-to-zero or a zero-to-one transition. Writing a zero to the appropriate edge bit of the active edge

(a) GPIP Data Register (GPIP)

Address 01 (Hex)	7	6	5	4	3	2	1	0
	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

SET a) MPU writes a one
CLEARED a) MPU writes a zero

(b) Active Edge Register (AER)

Address 03 (Hex)	7	6	5	4	3	2	1	0
	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, interrupts will be generated on the falling edge of the associated input signal. When a bit is a one, interrupts will be generated on the rising edge of the associated input signal.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

(c) Data Direction Register (DDR)

Address 05 (Hex)	7	6	5	4	3	2	1	0
	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, the associated I/O line is defined to be an input. When a bit is a one, the associated I/O line is defined to be an output.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

Figure 4-1. GPIP Control Registers

register causes the associated input to generate an interrupt on the one-to-zero transition. Writing a one to the edge bit will produce an interrupt on the zero-to-one transition of the corresponding GPIP line.

Note: The transition detector is an exclusive-OR gate whose inputs are the edge bit and the input buffer. As a result, writing the AER may cause an interrupt-producing transition, depending upon the state of the input. So, the AER should be configured before enabling interrupts via the interrupt enable registers (IERA and IERB). Also, changing the edge bit while interrupts are enabled may cause an interrupt on the corresponding channel.

4.2.3 Data Direction Register

The data direction register (DDR) allows the programmer to define I0 through I7 as inputs or outputs by writing the corresponding bit. When a bit of the data direction register is written as a zero, the corresponding interrupt I/O pin will be a high-impedance input. Writing a one to any bit of the data direction register will cause the corresponding pin to be configured as a push-pull output.

SECTION 5
TIMERS

The CMFP contains four 8-bit timers which provide many functions typically required in microprocessor systems. The timers can supply the baud rate clocks for the on-chip serial I/O channel, generate periodic interrupts, measure elapsed time, and count signal transitions. In addition, two timers have waveform generation capability.

All timers are prescaler/counter timers with a common independent clock input (XTAL1 or XTAL2) and are not required to be operated from the system clock. Each timer's output signal toggles when the timer's main counter times out. Additionally, timers A and B have auxiliary control signals which are used in two of the operation modes. An interrupt channel is assigned to each timer and when the auxiliary control signals are used, a separate interrupt channel will respond to transitions on these inputs.

5.1 OPERATION MODES

Timers A and B are full function timers which, in addition to the delay mode, operate in the pulse width measurement mode and the event count mode. Timers C and D are delay timers only. A brief discussion of each of the timer modes follows.

5.1.1 Delay Mode Operation

All timers may operate in the delay mode. In this mode, the prescaler is always active. The prescaler specifies the number of timer clock cycles which must elapse before a count pulse is applied to the main counter. A count pulse causes the main counter to decrement by one. When the timer has decremented down to 01 (hexadecimal), the next count pulse will cause the main counter to be reloaded from the timer data register and a time out pulse will be produced. This time out pulse is coupled to the timer's interrupt channel and, if the channel is enabled, an interrupt will occur. The time out pulse also causes the timer output pin to toggle. The output will remain in this new state until the next time out pulse occurs.

For example, if delay mode with a divide-by-10 prescaler is selected and the timer data register is loaded with 100 (decimal), the main counter will decrement once every 10 timer clock cycles. After 1,000 timer clocks, a time out pulse will be produced. This time out pulse will generate an interrupt if the channel is enabled (IERA, IERB) and in addition, the timer's output line will toggle. The output line will complete one full period every 2,000 cycles of the timer clock.

If the prescaler value is changed while the timer is enabled, the first time out pulse will occur at an indeterminate time no less than one nor more than 200 timer clock cycles. Subsequent time out pulses will then occur at the correct interval.

If the main counter is loaded with 01 (hexadecimal), a time out pulse will occur every time th prescaler presents a count pulse to the main counter. If the main counter is loaded with 00, a tim out pulse will occur every 256 count pulses.

5.1.2 Pulse Width Measurement Operation

Besides the delay mode, timers A and B may be programmed to operate in the pulse width mea surement mode. In this mode an auxiliary control input is required; timers A and B auxiliary inpu lines are TAI and TBI. Also, in the pulse width measurement mode, interrupt channels normall associated with I4 and I3 will respond to transitions on TAI and TBI, respectively. General purpos lines I3 and I4 may still be used for I/O. A conceptual circuit of the timers in the pulse width mea surement mode is shown in Figure 5-1.

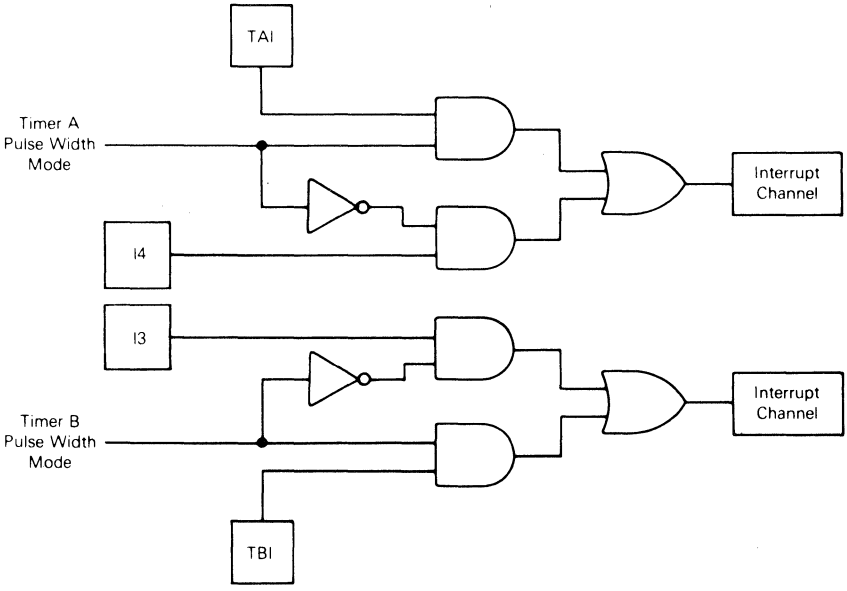


Figure 5-1. Conceptual Circuit of Timers A and B in Pulse Width Measurement Mode

The pulse width measurement mode functions similarly to the delay mode, with the auxiliary contro signal acting as an enable to the timer. When the control signal is active, the prescaler and mai counter are allowed to operate. When the control signal is negated, the timer is stopped. So, th width of the active pulse on TAI or TBI is measured by the number of timer counts which occu while the timer is allowed to operate.

The active state of the auxiliary input line is defined by the associated interrupt channel's edge bit i the active edge register (AER). GPIP4 of the AER is the edge bit associated with TAI and GPIP3 associated with TBI. When the edge bit is a one, the auxiliary input will be active high, enabling th timer while the input signal is at a high level. If the edge bit is low, the auxiliary input will be activ low and the timer will operate while the input signal is at a low level.

The state of the active edge bit also specifies whether a zero-to-one transition or a one-to-zero transition of the auxiliary input pin will produce an interrupt when the interrupt channel is enabled. In normal operation, programming the active edge bit to a one will produce an interrupt on the zero-to-one transition of the associated input signal. Alternately, programming the edge bit to a zero will produce an interrupt on the one-to-zero transition of the input signal. However, in the pulse width measurement mode, the interrupt generated by a transition on TAI or TBI will occur on the opposite transition as that normally defined by the edge bit.

For example, in the pulse width measurement mode, if the edge bit is a one, the timer will be allowed to run while the auxiliary input TAI is high. When TAI transitions from high to low, the timer will stop and, if the interrupt channel is enabled, an interrupt will occur. By having the interrupt occur on the one-to-zero transition instead of the zero-to-one transition, the processor will be interrupted when the pulse being measured has terminated and the width of the pulse is available from the timer. Therefore, the timers act like a divide-by-prescaler that can be programmed by the timer data register and the timers' A and B control register.

After reading the contents of the timer, the main counter must be reinitialized by writing to the timer data register to allow consecutive pulses to be measured. If the timer is written after the auxiliary input signal is active, the timer will count from the previous contents of the timer data register until it counts through 01 (hexadecimal). At that time, the main counter is loaded with the new value from the timer data register, a time out pulse is generated which will toggle the timer output, and an interrupt may be optionally generated on the timer interrupt channel. Note that the pulse width measured will include counts from before the main counter was reloaded. If the timer data register is written while the pulse is transitioning to the active state, an indeterminate value may be written into the main counter.

Once the timer is reprogrammed for another mode, interrupts will again occur as normally defined by the edge bit. Note that an interrupt may be generated as the result of placing the timer into the pulse width measurement mode or by reprogramming the timer for another mode. Also, an interrupt may be generated by changing the state of the edge bit while in the pulse width measurement mode.

.1.3 Event Count Mode Operation

In addition to the delay mode and the pulse width measurement mode, timers A and B may be programmed to operate in the event count mode. Like the pulse width measurement mode, the event count mode also requires an auxiliary input signal, TAI or TBI, and the interrupt channels normally associated with I4 and I3 will respond to transitions on TAI and TBI, respectively. General purpose lines I3 and I4 still function normally.

In the event count mode the prescaler is disabled, allowing each active transition on TAI and TBI to produce a count pulse. The count pulse causes the main counter to decrement by one. When the timer counts through 01 (hexadecimal), a time out pulse is generated which will cause the output signal to toggle and may optionally produce an interrupt via the associated timer interrupt channel. The timer's main counter is also reloaded from the timer data register. To count transitions reliably, the input signal may only transition once every four timer clock periods. For this reason, the input signal must have a maximum frequency equal to one-fourth that of the timer clock.

The active edge of the auxiliary input signal is defined by the associated interrupt channel's ed bit. GPIP4 of the AER specifies the active edge for TAI and GPIP3 defines the active edge for TE When the edge bit is programmed to a one, a count pulse will be generated on the zero-to-one tra sition of the auxiliary input signal. When the edge bit is programmed to a zero, a count pulse will generated on the one-to-zero transition. Also, note that changing the state of the edge bit while t timer is in the event count mode may produce a count pulse.

Besides generating a count pulse, the active transition of the auxiliary input signal will also produ an interrupt on the I3 or I4 interrupt channel, if the interrupt channel is enabled. Typically, in t event count mode, these channels are not enabled since the timer is automatically counting tran tions on the input signal. If the interrupt channel were enabled, the number of transitions could counted in the interrupt routine without requiring the use of the timer.

5.2 TIMER REGISTERS

The four timers are programmed via three control registers and four timer data registers. Contr registers TACR and TBCR and timer data registers TADR and TBDR (refer to Figure 5-1) a associated with timers A and B respectively. Timers C and D are controlled by the control regist TCDCR and the data registers TCDR and TDDR (refer to Figure 5-2).

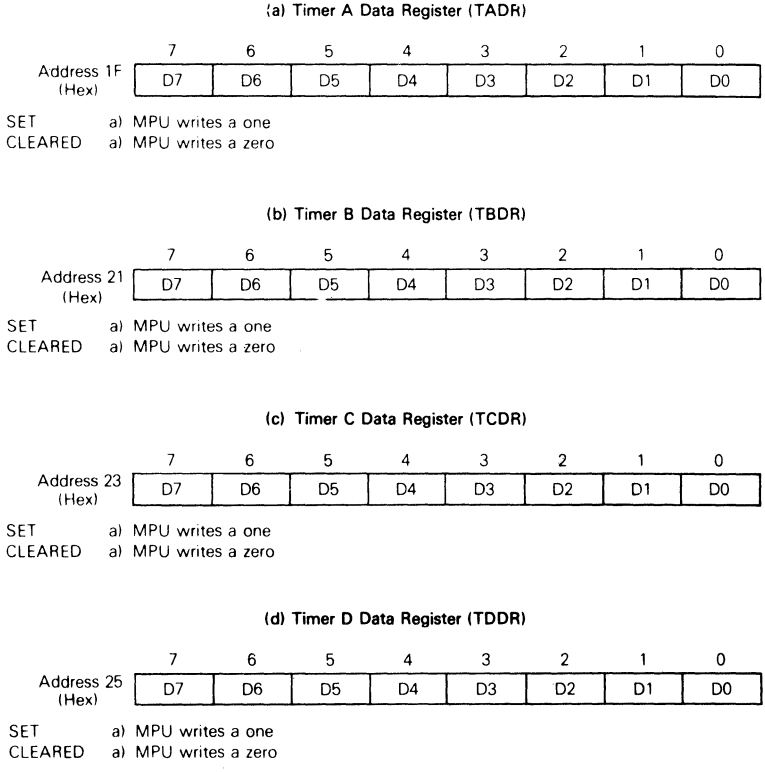


Figure 5-2. Timer Data Registers

5.2.1 Timer Data Registers

Each timer's main counter is an 8-bit binary down counter. The value of the main counter may be read at any time by reading the timer's data register. The information read is the value of the counter which was captured on the last low-to-high transition of the $\overline{DS}$ pin.

The main counter is initialized by writing to the timer's data register. If the timer is stopped, data is loaded simultaneously into both the timer data register and the main counter. If the timer data register is written while the timer is enabled, the value is not loaded into the timer until the timer counts through 01 (hexadecimal). Writing the timer data register while the timer is counting through 01 (hexadecimal) will cause an indeterminate value to be loaded into the timer's main counter. The four data registers are shown in Figure 5-2.

5.2.2 Timer Control Registers

Bits in the timer control registers select the operation mode, select the prescale value, and disable the timers. Timer control registers TACR and TBCR also have bits which allow the programmer to reset output lines TAO and TBO. These control registers are shown in Figure 5-3.

(a) Timer A Control Register (TACR)

	7	6	5	4	3	2	1	0
Address 19 (Hex)	*	*	*	Reset TA0	AC3	AC2	AC1	AC0

*Unused bits read as zero

(b) Timer B Control Register (TBCR)

	7	6	5	4	3	2	1	0
Address 1B (Hex)	*	*	*	Reset TB0	BC3	BC2	BC1	BC0

*Unused bits read as zero

Reset TAO/TBO — Timer's A and B output lines (TAO and TBO) may be forced low at any time by writing a one to the reset location in TACR and TBCR, respectively. The output will be held low only during the write operation; at the conclusion of the operation, the output will be allowed to toggle in response to a time-out pulse. When resetting TAO and TBO, the remaining bits in the control register must be written with their previous value to avoid altering the operating mode.

SET — a) End of write cycle which clears the bit

CLEARED — a) MPU writes a zero
b) Reset

AC3-AC0, BC3-BC0 These bits are decoded to determine the timer operation mode.

Figure 5-3. Timer Control Registers (Sheet 1 of 2)

AC3 BC3	AC2 BC2	AC1 BC1	AC0 BC0	Operation Mode
0	0	0	0	Timer Stopped*
0	0	0	1	Delay Mode, ÷ 4 Prescaler
0	0	1	0	Delay Mode, ÷ 10 Prescaler
0	0	1	1	Delay Mode, ÷ 16 Prescaler
0	1	0	0	Delay Mode, ÷ 50 Prescaler
0	1	0	1	Delay Mode, ÷ 64 Prescaler
0	1	1	0	Delay Mode, ÷ 100 Prescaler
0	1	1	1	Delay Mode, ÷ 200 Prescaler
1	0	0	0	Event Count Mode
1	0	0	1	Pulse Width Mode, ÷ 4 Prescaler
1	0	1	0	Pulse Width Mode, ÷ 10 Prescaler
1	0	1	1	Pulse Width Mode, ÷ 16 Prescaler
1	1	0	0	Pulse Width Mode, ÷ 50 Prescaler
1	1	0	1	Pulse Width Mode, ÷ 64 Prescaler
1	1	1	0	Pulse Width Mode, ÷ 100 Prescaler
1	1	1	1	Pulse Width Mode, ÷ 200 Prescaler

*Regardless of the operation mode, counting is inhibited when the timer is stopped. The contents of the timer's main counter is not affected, although any residual count in the prescaler is lost.

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

(c) Timers C and D Control Register (TCDCR)

	7	6	5	4	3	2	1	0
Address 1D (Hex)	*	CC2	CC1	CC0	*	DC2	DC1	DC0

*Unused bits read as zero

CC2-CC0, DC2-DC0 The bits are decoded to determine the timer operation mode.

CC2 DC2	CC1 DC1	CC0 DC0	Operation Mode
0	0	0	Timer Stopped*
0	0	1	Delay Mode, ÷ 4 Prescaler
0	1	0	Delay Mode, ÷ 10 Prescaler
0	1	1	Delay Mode, ÷ 16 Prescaler
1	0	0	Delay Mode, ÷ 50 Prescaler
1	0	1	Delay Mode, ÷ 64 Prescaler
1	1	0	Delay Mode, ÷ 100 Prescaler
1	1	1	Delay Mode, ÷ 200 Prescaler

*When the timer is stopped, counting is inhibited. The contents of the timer's main counter is not affected, although any residual count in the prescaler is lost.

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

Figure 5-3. Timer Control Registers (Sheet 2 of 2)

SECTION 6
UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER-TRANSMITTER

The universal synchronous/asynchronous receiver-transmitter (USART) is a single full-duplex serial channel with a double-buffered receiver and transmitter. There are separate receive and transmit clocks and separate receive and transmit status and data bytes. The receive and transmit sections are also assigned separate interrupt channels. Each section has both a normal condition interrupt channel and an error condition interrupt channel. These channels can be optionally disabled from interrupting the processor and instead, DMA transfers can be performed using the receiver ready and transmitter ready external CMFP signals.

6.1 CHARACTER PROTOCOLS

The CMFP USART supports asynchronous and with the aid of a polynomial generator checker (PGC) supports byte synchronous character formats. These formats are selected independently of the divide-by-one and divide-by-16 clock modes.

When the divide-by-one clock mode is selected, synchronization must be accomplished externally. The receiver will sample the serial data on the rising edge of the receiver clock. In the divide-by-16 clock mode, the data is sampled at mid-bit time to increase transient noise rejection.

Also, when the divide-by-16 clock mode is selected, the USART resynchronization logic is enabled. This logic increases the channel's clock skew tolerance. When a valid transition is detected, an internal counter is reset to state zero. Transition checking is then inhibited until state four. Then at state eight, the previous state of the transition checking logic is clocked into the receive shift register.

6.1.1 Asynchronous Format

Variable word length and start/stop bit configurations are available under software control for asynchronous operation. The word length can be five to eight bits and one, one and one-half, or two stop bits can be selected. The user can also select odd, even, or no parity. For character lengths of less than eight bits, the assembled character will consist of the required number of data bits followed by zeros in the unused bit positions and a parity bit, if parity is enabled.

In the asynchronous format, start bit detection is always enabled. New data is not shifted into the receive shift register until a zero bit is received. When the divide-by-16 clock mode is selected, the false start bit logic is also active. Any transition must be stable for three positive receive clock edges to be considered valid. Then a valid zero-to-one transition must not occur for at least eight additional positive clock edges.

6.1.1.1 WAKE-UP FEATURE. In a typical serial loop multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, wake-up feature is included whereby all further USART receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An USART receiver is re-enabled by an idle string of ten consecutive ones or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

6.1.2 Synchronous Format

When the synchronous character format is selected, the 8-bit synchronous character loaded in the synchronous character register is compared to received serial data until a match is found. On synchronization is established, incoming data is clocked into the receiver. The synchronous wo will be continuously transmitted during an underrun condition. All synchronous characters can optionally stripped from the receive buffer. Figure 6-1 shows the synchronous character registe

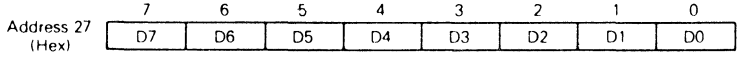


Figure 6-1. Synchronous Character Register (SCR)

The synchronous character is typically written after the data word length is selected, sinc unused bits in the synchronous character register are zeroed out. When parity is enable synchronous word length is the data word length plus one. The CMFP will compute and apper the parity bit for the synchronous word when a word length of eight is selected. However, if th word length is less than eight, the user must determine the synchronous word parity and wri it into the synchronous character register along with the synchronous character. The CMFP w then transmit the extra bit in the synchronous word as a parity bit.

6.1.3 USART Control Register

The USART control register (UCR) selects the clock mode and the character format for the recei and transmit sections. This register is shown in Figure 6-2.

6.2 RECEIVER

As data is received on the serial input line (SI), it is clocked into an internal 8-bit shift register un the specified number of data bits have been assembled. This character will then be transferred the receive buffer, assuming that the last word in the receiver buffer has been read. This transf produces a buffer full interrupt to the processor.

Reading the receive buffer satisfies the buffer full condition and allows a new data word to transferred to the receive buffer when it is assembled. The receive buffer is accessed by reading t USART data register (UDR). The UDR is simply an 8-bit data register used when transferring da from the CMFP and the CPU.

Each time a word is transferred to the receive buffer, its status information is latched into t receiver status register (RSR). The RSR is not updated again until the data word in the receive bu fer has been read. When a buffer full condition exists, the RSR should always be read before t receive buffer (UDR) to maintain the correct correspondance between data and flags. Otherwise, is possible that after reading the UDR and prior to reading the RSR, a new word could be receiv and transferred to the receive buffer. Its associated flags would be latched into the RSR, ove writing the flags for the previous data word. Then when the RSR were read to access the stat information for the first data word, the flags for the new word would be retrieved.

Address 29 (Hex)	7	6	5	4	3	2	1	0
	CLK	WL1	WL0	ST1	ST0	PE	E/O	WU

CLK — Clock Mode. When this bit is zero, data will be clocked into and out of the receiver and transmitter at the frequency of their respective clocks. When this bit is a one, data will be clocked into and out of the receiver and transmitter at one sixteenth the frequency of their respective clocks. Also, the receiver data transition resynchronization logic will be enabled.

SET = ÷16 a) MPU writes a one
CLEARED = ÷1 a) MPU writes a zero
b) Reset

WL0, WL1 — Word Length. These two bits specify the length of the data word exclusive of start bits, stop bits, and parity.

WL1	WL0	Word Length
0	0	8 Bits
0	1	7 Bits
1	0	6 Bits
1	1	5 Bits

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

ST0, ST1 — Start/Stop Bit and Format Control. These two bits select the number of start and stop bits and also specify the character format.

ST1	ST0	Start Bits	Stop Bits	Format
0	0	0	0	Synchronous
0	1	1	1	Asynchronous
1	0	1	1½	Asynchronous*
1	1	1	2	Asynchronous

*Only used with divide-by-16 clock mode

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

PE — Parity Enable. When this bit is zero, no parity check will be made and no parity bit will be computed for transmission. When this bit is a one, parity will be checked by the receiver and parity will be calculated and inserted during data transmission. Note that parity is not automatically appended to the synchronous character for word lengths of less than eight bits. In this case, the parity should be written into the synchronous character register along with the synchronous word.

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

E/O — Even/Odd Parity. When this bit is zero, odd parity is selected. When this bit is a one, even parity is selected.

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

WU — Bit 0 "Wake-up" on idle line. When set, WU enables the wake-up function; it is cleared by ten consecutive ones or during reset. WU will not be set if the line is idle. Refer to 6.1.1.1.

Figure 6-2. USART Control Register (UCR)

6.2.1 Receiver Interrupt Channels

The USART receive section is assigned two interrupt channels. One indicates the buffer full cond tion, while the other channel indicates an error condition. Error conditions include overrun, parit error, synchronous found, and break. These interrupting conditions correspond to the BF, OE, PE and F/S or B bits of the receiver status register. These flags will function as described in **6.2.** whether the receiver interrupt channels are enabled or disabled.

While only one interrupt is generated per character received, two dedicated interrupt channels allo separate vector numbers to be assigned for normal and abnormal receiver conditions. When received word has an error associated with it and the error interrupt channel is enabled, an interru will be generated on the error channel only. However, if the error channel is disabled, an interru for an error condition will be generated on the buffer full interrupt channel along with interrupts pr duced by the buffer full condition. The receiver status register must always be read to determin which error condition produced the interrupt.

6.2.2 Receiver Status Register

The receiver status register contains the receive buffer full flag, the synchronous strip enable, th receiver enable, and various status information associated with the data word in the receive buffe The RSR is latched each time a data word is transferred to the receive buffer. RSR flags canno change again until the data word has been read. The exception is the character in progress fla which monitors when a new word is being assembled in the asynchronous character format. Th receiver status register is shown in Figure 6-3.

Address 2B	BF	OE	PE	FE	F/S or B	M/CIP	SS	RE

BF — Buffer Full. This bit is set when a received word is transferred to the receive buffer. This bit is cleared when the receive buffer is read by accessing the USART data register (UDR). This bit is read only.
SET a) Received word transferred to buffer
CLEARED a) Receive buffer read
b) Reset

OE — Overrun Error. An overrun error occurs when a received word is due to be transferred to the receive buffer, but the receive buffer is full. Neither the receive buffer nor the RSR is overwritten. The OE bit is set after the receive buffer full condition is satisfied by reading the UDR. This error condition will generate an interrupt to the processor. The OE bit is cleared by reading the RSR. New data words will not be assembled until the RSR is read.
SET a) Incoming word received and receive buffer full
CLEARED a) Receiver status register read
b) Reset

PE — Parity Error. This bit is set when the word transferred to the receive buffer has a parity error. This bit is cleared when the word transferred to the receive buffer does not have a parity error.
SET a) Word in receive buffer has a parity error
CLEARED a) Word in receive buffer does not have a parity error
b) Reset

FE — Frame Error. A frame error exists when a non-zero data word is not followed by a stop bit in the asynchronous character format. The FE bit is set when the word transferred to the receive buffer has a frame error. The FE bit is cleared when the word transferred to the receive buffer does not have a frame error.
SET a) Word in receive buffer has a frame error
CLEARED a) Word in receive buffer does not have a frame error
b) Reset

Figure 6-3. Receiver Status Register (RSR) (Sheet 1 of 2)

F/S or B — Found/Search or Break Detect. In the synchronous character format this bit can be set or cleared in software. When the bit is a zero, the USART receiver is placed in the search mode. The incoming data is compared to the synchronous character register (SCR) and the word length counter is disabled. The F/S bit will automatically be set when a match is found and the word length counter will be enabled. An interrupt will also be produced on the receive error channel.

SET
a) Incoming word matches synchronous character

CLEARED
a) MPU writes a zero
b) Incoming word does not match synchronous character
c) Reset

In the asynchronous character format, this flag indicates a break condition. A break is detected when an all zero data word with no stop bit is received. The break condition continues until a non-zero data bit is received. The B bit is set when the word transferred to the receive buffer is a break indication. A break condition generates an interrupt to the processor. This bit is cleared when a non-zero data bit is received and the break condition has been acknowledged by reading the RSR at least once. An end of break interrupt will be generated when the bit is cleared.

SET
a) Word in receive buffer is a break

CLEARED
a) Break terminates and receiver status register read since beginning of break condition
b) Reset

M or CIP — Match/Character in Progress. In the synchronous character format, this flag indicates that a synchronous character has been received. The M bit is set when the word transferred to the receive buffer matches the synchronous character register. The M bit is cleared when the word transferred to the receive buffer does not match the synchronous character register.

SET
a) Word transferred to receive buffer matches the synchronous character

CLEARED
a) Word transferred to receive buffer does not match synchronous character
b) Reset

In the asynchronous character format, this flag indicates that a word is being assembled. The CIP bit is set when a start bit is detected. The CIP bit is cleared when the final stop bit has been received.

SET
a) Start bit is detected

CLEARED
a) End of word detected
b) Reset

SS — Synchronous Strip Enable. When this bit is a one, data words that match the synchronous character register will not be loaded into the receive buffer and no buffer full condition will be produced. When this bit is a zero, data words that match the synchronous character register will be transferred to the receive buffer and a buffer full condition will be produced.

SET
a) MPU writes a one

CLEARED
a) MPU writes a zero
b) Reset

RE — Receiver Enable. When this bit is a zero, the receiver will be immediately disabled. All flags will be cleared. When this bit is a one, normal receiver operation is enabled. This bit should not be set to a one until the receiver clock is active.

SET
a) MPU writes a one
b) Transmitter is disabled in auto-turnaround mode

CLEARED
a) MPU writes a zero
b) Reset

Figure 6-3. Receiver Status Register (RSR) (Sheet 2 of 2)

6.2.3 Special Receive Considerations

Certain receive conditions relating to the overrun error flag and the break detect flag require further explanation. Consider the following examples:

1) A break is received while the receive buffer is full.
   This does not produce an overrun condition. Only the B flag will be set after the receiver buffer is read.
2) A new word is received and the receive buffer is full. A break is received before the receive buffer is read.
   Both the B and OE flags will be set when the buffer full condition is satisfied.

6.3 TRANSMITTER

The transmit buffer is loaded by writing to the USART data register (UDR). The data word will b transferred to an internal 8-bit shift register when the last word in the shift register has bee transmitted. This will produce a buffer empty condition. If the transmitter completes the transmis sion of the word in the shift register before a new word is written to the transmit buffer, an underru error will occur. In the asynchronous character format, the transmitter will send a mark until th transmit buffer is written. In the synchronous character format, the transmitter will continuousl send the synchronous character.

The transmit buffer can be loaded prior to enabling the transmitter. After the transmitter is enabled there is a delay before the first bit is output. The serial output line (SO) should be programmed to b high, low, or high impedance when the transmitter is enabled to force the output line to the desire state until the first bit is shifted out. Note that a one bit will always be transmitted prior to the wor in the transmit shift register when the transmitter is first enabled.

When the transmitter is disabled, any word currently being transmitted will continue to completion However, any word in the transmit buffer will not be transmitted and will remain in the buffer. So no buffer empty condition will occur. If the buffer is empty when the transmitter is disabled, th buffer empty condition will remain, but no underrun condition will be generated when the word i transmission is completed. If no word is being transmitted when the transmitter is disabled, th transmitter will stop at the next rising edge of the internal shift clock.

In the asynchronous character format, the transmitter can be programmed to send a break. Th break will be transmitted once the word currently in the shift register has been sent. If the shif register is empty, the break command will be effective immediately. An END interrupt will b generated at every normal character boundary to aid in timing the break transmission. The brea will continue until the break command is cleared.

Any character in the transmit buffer at the start of a break will be transmitted when the break is ter minated. If the transmit buffer is empty at the start of a break, it may be written at any time durin the break. If the buffer is still empty at the end of the break, an underrun condition will exist.

Disabling the transmitter during a break condition causes the transmitter to cease transmission o the break character at the end of the current character. No end of break stop bit will be transmitted Even if the transmit buffer is empty, no buffer empty condition will occur nor will an underrun con dition occur. Also, any word in the transmit buffer will remain.

6.3.1 Transmitter Interrupt Channels

The USART transmit section is assigned two interrupt channels. One channel indicates a buffe empty condition and the other channel indicates an underrun or end condition. These interruptin conditions correspond to the BE, UE, and END flag bits of the transmitter status register (TSR). Th **flag bits will function as described in 6.3.2 whether their associated interrupt channel i enabled or disabled.**

6.3.2 Transmitter Status Register

The transmitter status register contains various transmitter error flags and transmitter control bit for selecting auto-turnaround and loopback mode. The TSR is shown in Figure 6-4.

	7	6	5	4	3	2	1	0
Address 2D (Hex)	BE	UE	AT	END	B	H	L	TE

BE — Buffer Empty. This bit is set when the word in the transmit buffer is transferred to the transmit shift register. This bit is cleared when the transmit buffer is reloaded by writing to the USART data register (UDR).
SET a) Transmit buffer contents transferred to transmit shift register
CLEARED a) Transmit buffer written

UE — Underrun Error. This bit is set when the word in the transmit shift register has been transmitted before a new word is loaded into the transmit buffer. This bit is cleared by reading the TSR or by disabling the transmitter. This bit does not need to be cleared before writing to the UDR.
SET a) Transmit shift register contents transmitted before transmit buffer written
CLEARED a) Transmitter status register read
b) Transmitter disabled

AT — Auto-Turnaround. When this bit is set, the receiver will be enabled automatically after the transmitter has been disabled and the last character being transmitted is completed.
SET a) MPU writes a one
CLEARED a) Transmitter disabled

END — End of Transmission. When the transmitter is disabled while a character is being transmitted, the END will be set after the character transmission is complete. If no word is being transmitted when the transmitter is disabled, the END bit will be set immediately. The END bit is cleared by reenabling the transmitter.
SET a) Transmitter disabled
CLEARED a) Transmitter enabled

B — Break. This bit has no function in the synchronous character format. In the asynchronous character format, when this bit is set to a one, a break will be transmitted upon the completion of the transmission of any word in the transmit shift register. A break consists of an all zero data word with no stop bit. When this bit is cleared by software, the break indication will cease and normal transmission will resume. Note that when B is set, BE cannot be set.
SET a) MPU writes a one
CLEARED a) MPU writes a zero

H, L — High and Low. These control bits configure the transmitter output (SO) when the transmitter is disabled. These bits also force the transmitter output after the transmitter is enabled until END is cleared.

H	L	Output State
0	0	High Impedance
0	1	Low
1	0	High
1	1	Loopback Mode

Loopback mode internally connects the transmitter output to the receiver input and the transmitter clock to the receiver clock internally. The receiver clock (RC) and the serial input (SI) are not used. When the transmitter is disabled, SO is forced high.
SET a) MPU writes a one
CLEARED a) MPU writes a zero

TE — Transmitter Enable. When this bit is cleared, the transmitter is disabled. The UE bit will be cleared and the END bit will be set. When this bit is set, the transmitter is enabled. The transmitter output will be driven according to the H and L bits until transmission begins. A one bit will be transmitted before the transmission of the word in the transmit shift register is begun.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

Figure 6-4. Transmitter Status Register (TSR)

6.4 DMA OPERATION

USART error conditions are only valid for each character boundary. When the USART perform block data transfers by using the DMA handshake lines $\overline{RR}$ (receiver ready) and $\overline{TR}$ (transmitte ready), errors must be saved and checked at the end of a block. This is accomplished by enablin the error channel for the receiver or transmitter and by masking interrupts for this channel. Once th transfer is complete, interrupt pending register A is read. Any pending receiver or transmitter erro indicates an error in the data transfer.

SECTION 7
ELECTRICAL CHARACTERISTICS

This section contains the electrical specifications and associated timing information for the TS68HC901 multi-function peripheral.

7.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to 7.0	V
Input Voltage	V_{in}	−0.3 to 7.0	V
Operating Temperature Range TS68HC901C TS68HC901V TS68HC901M	T_A	T_L to T_H 0 to +70 −40 to +85 −55 to +125	°C
Storage Temperature	T_{stg}	−65 to 150	°C
Power Dissipation	P_D	30	mW

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{CC} or GND).

7.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance Ceramic Plastic	θ_{JA}	 40 TBD	 °C/W

7.3 POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} P_D^2 \quad (3)$$

Where:

K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

7.4 DC ELECTRICAL CHARACTERISTICS

($T_A = T_L$ to T_H $V_{CC} = -5$ V ±5%, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Input High Voltage except XTAL1, XTAL2	V_{IH}	2.0	$V_{DD}+0.3$	V
Input High Voltage XTAL1, XTAL2	V_{IH}	$V_{DD}-1.5$	$V_{DD}+0.3$	V
Input Low Voltage	V_{IL}	−0.3	0.8	V
Output High Voltage, Except $\overline{DTACK}$ ($I_{OH} = -120\ \mu A$)	V_{OH}	4.1	–	V
Output Low Voltage, Except $\overline{DTACK}$ ($I_{OL} = 2.0$ mA)	V_{OL}	–	0.5	V
Power Supply Current (Outputs Open)	I_{LL}	–	3.7	mA
Input Leakage Current ($V_{in} = 0$ to V_{CC})	I_{LI}	–	±10	μA
Hi-Z Output Leakage Current in Float ($V_{out} = 2.4$ to V_{CC})	I_{LOH}	–	10	μA
Hi-Z Output Leakage Current in Float ($V_{out} = 0.5$ V)	I_{LOL}	–	−10	μA
$\overline{DTACK}$ Output Source Current ($V_{out} = 2.4$ V)	I_{OH}	–	−400	μA
$\overline{DTACK}$ Output Sink Current ($V_{out} = 0.5$ V)	I_{OL}	–	5.3	mA
Pull Down Resistor	R_{MPX}	1	3	MΩ

7.5 CAPACITANCE ($T_A = 25°C$, f = 1 MHz, unmeasured pins returned to ground)

Characteristic	Symbol	Min	Max	Unit
Input Capacitance	C_{in}	–	10	pF
Hi-Z Output Capacitance	C_{out}	–	10	pF

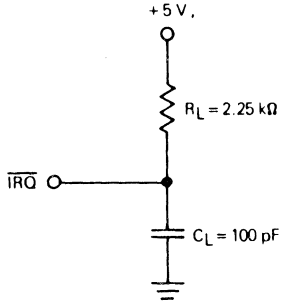
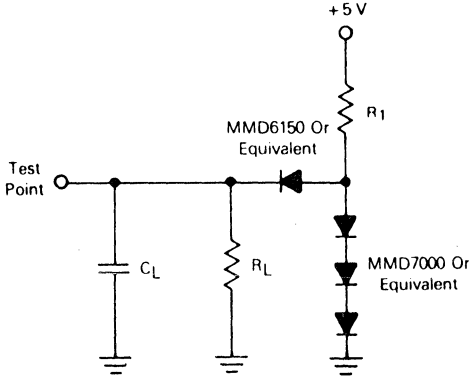


Figure 7-1. $\overline{IRQ}$ Test Load



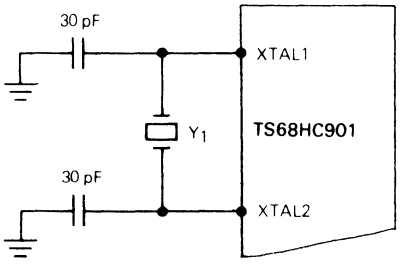
for all outputs except $\overline{DTACK}$
$C_L = 100$ pF
$R_L = 20$ kΩ
$R_1 = 1.90$ kΩ

for $\overline{DTACK}$
$C_L = 130$ pF
$R_L = 6$ kΩ
$R_1 = 740\ \Omega$

Figure 7-2. Typical Test Load

7.6 CLOCK TIMING

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f	1.0	4.0	MHz
Cycle Time	t_{cyc}	250	1000	ns
Clock Pulse Width	t_{CL}, t_{CH}	110	250	ns
Rise and Fall Times	t_{Cr}, t_{Cf}	–	15	ns



Crystal Parameters
Parallel resonance fundamental mode AT cut
$R_S \leq 150\ \Omega$ (f = 2.8 – 4.0 MHz)
$R_S \leq 300\ \Omega$ (f = 2.0 – 2.7 MHz)
C_L = 18 pF, C_M = 0.02 pF, C_R = 5 pF, L_M = 96 MHz
f (typical) = 2.4576 MHz

Figure 7-3. CMFP External Oscillator Components

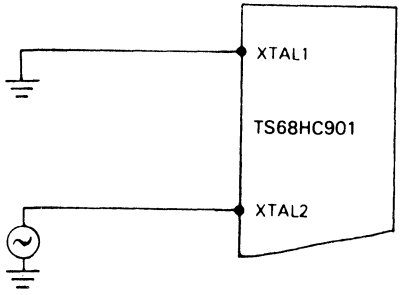


Figure 7-3-1. CMFP External Clock Connection

7.7 AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0V_{DC} ± 5%, V_{SS} = 0V_{DC}, T_A = 0°C to 70°C unless otherwise noted). See figures 7-4 through 7-10.

Num	Characteristic	4 MHz		8 MHz		Unit
		Min	Max	Min	Max	
1	$\overline{CS}$, $\overline{DS}$ Width High	50	—	50	—	—
2	R/$\overline{W}$, A1-A5 Valid to Falling $\overline{CS}$ (Setup)	30	—	20	—	ns
3	Data Valid Prior to Falling CLK	280	—	100	—	ns
4 (3)	$\overline{CS}$, $\overline{IACK}$ Valid to falling Clock (Setup)	50	—	50	—	ns
5	CLK Low to $\overline{DTACK}$ Low	—	220	—	90	ns
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ High	—	60	—	50	ns
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state	—	100	—	100	ns
8	$\overline{DTACK}$ Low to Data Invalid (Hold Time)	0	—	0	—	ns
9	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to Data Tri-state	—	50	—	50	ns
10	$\overline{CS}$ or $\overline{DS}$ High to R/$\overline{W}$, A1-A5 Invalid (Hold Time)	0	—	0	—	ns
11 (3,5)	Data Valid from $\overline{CS}$ Low	—	310	—	180	ns
12	Read Data Valid to $\overline{DTACK}$ Low (Setup Time)	50	—	0	—	ns
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (Hold Time)	0	—	0	—	ns
14	$\overline{IEI}$ Low to Falling $\overline{CLK}$ (Setup)	50	—	50	—	ns
15 (1)	$\overline{IEO}$ Valid from Clock Low (Delay)	—	180	—	120	ns
16	Data Valid from Clock Low (Delay)	—	300	—	180	ns
17	$\overline{IEO}$ Invalid from $\overline{IACK}$ High (Delay)	—	150	—	100	ns
18	$\overline{DTACK}$ Low from Clock High (Delay)	—	180	—	100	ns
19 (1)	$\overline{IEO}$ Valid from $\overline{IEI}$ Low (Delay)	—	100	—	100	ns
20	Data Valid from $\overline{IEI}$ Low (Delay)	—	220	—	140	ns
21	Clock Cycle Time	250	1000	125	1000	ns
22	Clock Width Low	110	—	55	—	ns
23	Clock Width High	110	—	55	—	ns
24 (4)	$\overline{CS}$, $\overline{IACK}$ Inactive to Rising Clock (Setup)	100	—	50	—	ns
25	I/O Minimum Active Pulse Width	100	—	100	—	ns
26	$\overline{IACK}$ Width High/Minimum Delay between two Pulses	2	—	2	—	CLK
27	I/O Data Valid from Rising $\overline{CS}$ or $\overline{DS}$	—	450	—	350	ns
28	Receiver Ready Delay from Falling RC	—	600	—	200	ns
29	Transmitter Ready Delay from Falling TC	—	600	—	200	ns
30 (6)	Timer Output Low from Rising Edge of $\overline{CS}$ or $\overline{DS}$ (A & B) (Reset T_{OUT})	—	450	—	200	ns

Num	Characteristic	4 MHz		8 MHz		Unit
		Min	Max	Min	Max	
31 (2)	T_{OUT} Valid from Internal Timeout	—	2 t_{CLK} + 300	—	2 t_{CLK} + 300	ns
32	Timer Clock Low Time	110	—	55	—	ns
33	Timer Clock High Time	110	—	55	—	ns
34	Timer Clock Cycle Time	250	1000	125	1000	ns
35	$\overline{RESET}$ Low Time	2	—	1.5	—	μs
36	Delay to Falling $\overline{INTR}$ from External Interrupt Active Transition	—	380	—	250	ns
37	Transmitter Internal Interrupt Delay from Falling Edge of TC	550	—	250	—	ns
38	Receiver Buffer Full Interrupt Transition Delay from Rising Edge of RC	800	—	400	—	ns
39	Receiver Error Interrupt Transition Delay from Falling Edge of RC	800	—	400	—	ns
40	Serial In Set Up Time to Rising Edge of RC (Divide by one only)	80	—	50	—	ns
41	Data Hold Time from Rising Edge of RC (Divide by one only)	350	—	100	—	ns
42	Serial Output Data Valid from Falling Edge of TC (÷ 1)	—	440	—	200	ns
43	Transmitter Clock Low Time	500	—	250	—	ns
44	Transmitter Clock High Time	500	—	250	—	ns
45	Transmitter Clock Cycle Time	1.05	*	0.55	*	μs
46	Receiver Clock Low Time	500	—	250	—	ns
47	Receiver Clock High Time	500	—	250	—	ns
48	Receiver Clock Cycle Time	1.05	*	0.55	*	μs
49 (2)	$\overline{CS}$, $\overline{IACK}$, $\overline{DS}$ Width Low	—	80	—	80	T_{CLK}
50	Serial Output Data Valid from Falling Edge of TC (÷ 16)	—	490	—	200	ns
51	Cycle Time	1000	—	—	—	ns
52	Pulse Width, E High	430	—	—	—	ns
53	Pulse Width, E Low	450	—	—	—	ns
54	Address, R/$\overline{W}$ Setup Time Before E	80	—	—	—	ns
55	$\overline{CS}$ Setup Time Before E	80	—	—	—	ns
56	Address Hold Time	10	—	—	—	ns
57	$\overline{CS}$ Hold Time	10	—	—	—	ns
58	Output Data Delay Time (Read)	—	250	—	—	ns
59	Data Hold Time (Read)	0	100	—	—	ns

Num	Characteristic	4 MHz		8 MHz		Unit
		Min	Max	Min	Max	
60	Input Data Setup Time (Write)	280	—	—	—	ns
61	Data Hold Time (Write)	20	—	—	—	ns
62	Cycle Time	800	—	—	—	ns
63	Pulse Width $\overline{DS}$ Low or RD/$\overline{WR}$ High	350	—	—	—	ns
64	Pulse Width DS High or RD/$\overline{WR}$ Low	340	—	—	—	ns
65	Pulse Width AS/ALE High	100	—	—	—	ns
66	Delay AS Fall to $\overline{DS}$ Rise or ALE Fall to RD/$\overline{WR}$ Fall	30	—	—	—	ns
67	Delay $\overline{DS}$ or RD/$\overline{WR}$ Rise to AS/ALE Rise	30	—	—	—	ns
68	R/$\overline{W}$ Setup Time to $\overline{DS}$	100	—	—	—	ns
69	R/$\overline{W}$ Hold Time to $\overline{DS}$	10	—	—	—	ns
70	Address Setup Time to AS/ALE	20	—	—	—	ns
71	Address Hold Time to AS/ALE	20	—	—	—	ns
72	Data Setup Time to $\overline{DS}$ or WR (Write)	280	—	—	—	ns
73	Delay Data to $\overline{DS}$ or RD (Read)	—	250	—	—	ns
74	Data Hold Time to $\overline{DS}$ or WR (Write)	20	—	—	—	ns
75	Data Hold Time to $\overline{DS}$ or RD (Read)	0	100	—	—	ns
76	CE Setup Time to AS/ALE Fall	20	—	—	—	ns
77	CE Hold Time to $\overline{DS}$, RD or $\overline{WR}$	20	—	—	—	ns

Notes:

1. $\overline{IEO}$ only goes low if no acknowledgeable interrupt is pending. If $\overline{IEO}$ goes low, $\overline{DTACK}$ and the data bus remain tri-stated.

2. T_{CLK} refers to the clock applied to the CMFP CLK input pin. t_{CLK} refers to the timer clock signal, regardless of whether that signal comes from the XTAL1/XTAL2 crystal clock inputs or the TAI or TBI timer inputs.

3. If the setup time is not met, $\overline{CS}$ or $\overline{IACK}$ will not be recognized until the next falling CLK.

4. If the setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.

5. Althrough $\overline{CS}$ and $\overline{DTACK}$ are synchronized with the clock, the data out during a reed cycle is asynchronous to the clock, relying only on $\overline{CS}$ for timing.

6. Spec. 30 applies to timer outputs TAO and TBO only.

7.7.1 AC ELECTRICAL CHARACTERISTICS - READ CYCLES

($V_{CC} = 5.0 V_{DC} \pm 5\%$, $V_{SS} = 0 V_{DC}$, $T_A = T_L$ to T_H unless otherwise noted)

Num	Characteristic	4 MHz		8 MHz		Unit
		Min	Max	Min	Max	
1	$\overline{CS}$, $\overline{DS}$ Width High	50	—	50	—	—
2	R/$\overline{W}$, A1-A5 Valid to Falling $\overline{CS}$ (Setup)	30	—	20	—	ns
4 (3)	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (Setup)	50	—	50	—	ns
5	CLK Low to $\overline{DTACK}$ Low	—	220	—	90	ns
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ High	—	60	—	50	ns
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state	—	100	—	100	ns
8	$\overline{DTACK}$ Low to Data Invalid (Hold Time)	0	—	0	—	ns
9	$\overline{CS}$, $\overline{DS}$ or IACK High to Data Tri-state	—	50	—	50	ns
10	$\overline{CS}$ or $\overline{DS}$ High to R/$\overline{W}$, A1-A5 Invalid (Hold Time)	0	—	0	—	ns
11 (3,5)	Data Valid from $\overline{CS}$ Low	—	310	—	180	ns
12	Read Data Valid to $\overline{DTACK}$ Low (Setup Time)	50	—	0	—	ns
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (Hold Time)	0	—	0	—	ns
24 (4)	$\overline{CS}$, $\overline{IACK}$ Inactive to Rising Clock (Setup)	100	—	50	—	ns

Notes:

3. If the setup time is not met, $\overline{CS}$ or $\overline{IACK}$ will not be recognized until the next falling CLK.

4. If the setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.

5. Although $\overline{CS}$ and $\overline{DTACK}$ are synchronized with the clock, the data out during a reed cycle is asynchronous to the clock, relying only on $\overline{CS}$ for timing.

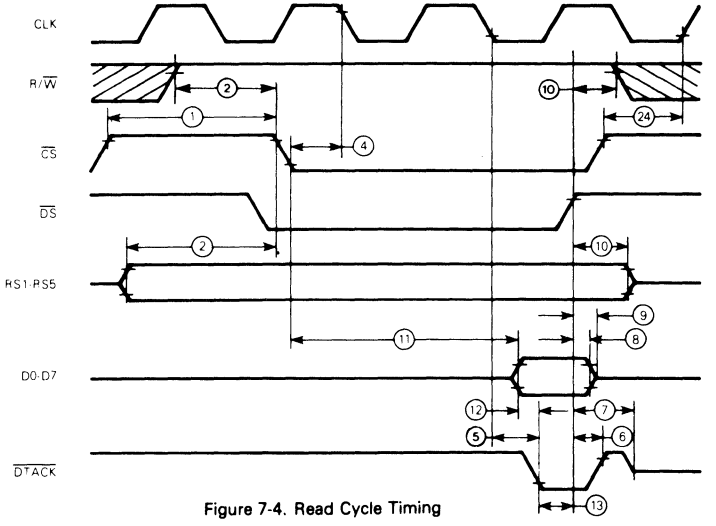


Figure 7-4. Read Cycle Timing

7.7.2 AC ELECTRICAL CHARACTERISTICS - WRITE CYCLES

(V_{CC} = 5.0 V_{DC} ± 5 %, V_{SS} = 0 V_{DC}, T_A = T_L to T_H unless otherwise noted)

Num	Characteristic	4 MHz		8 MHz		Unit
		Min	Max	Min	Max	
1	$\overline{CS}$, $\overline{DS}$ Width High	50	—	50	—	—
2	R/$\overline{W}$, A1-A5 Valid to Falling $\overline{CS}$ (Setup)	30	—	20	—	ns
3	Data Valid Prior to Falling CLK	280	—	100	—	ns
4 (3)	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (Setup)	50	—	50	—	ns
5	$\overline{CLK}$ Low to $\overline{DTACK}$ Low	—	220	—	90	ns
6	$\overline{CS}$, $\overline{DS}$ or IACK High to $\overline{DTACK}$ High	—	60	—	50	ns
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state	—	100	—	100	ns
8	$\overline{DTACK}$ Low to Data Invalid (Hold Time)	0	—	0	—	ns
10	$\overline{CS}$ or $\overline{DS}$ High to R/$\overline{W}$, A1-A5 Invalid (Hold Time)	0	—	0	—	ns
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (Hold Time)	0	—	0	—	ns
24 (4)	$\overline{CS}$, $\overline{IACK}$ Inactive to Rising Clock (Setup)	100	—	50	—	ns

Notes:

3. If the setup time is not met, $\overline{CS}$ or $\overline{IACK}$ will not be recognized until the next falling CLK.

4. If the setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If no
met, the hold-off will be two clock cycles.

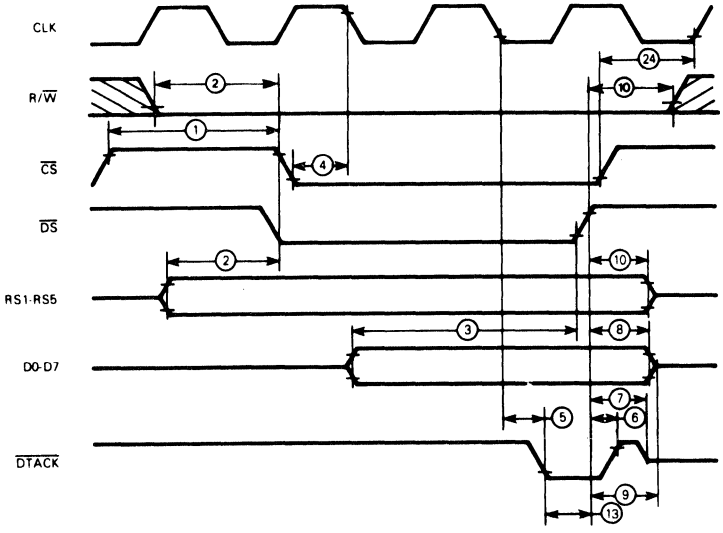


Figure 7-5. Write Cycle Timing

7.7.3 AC ELECTRICAL CHARACTERISTICS - INTERRUPT ACKNOWLEDGE CYCLES

($V_{CC} = 5.0 V_{DC} \pm 5\%$, $V_{SS} = 0 V_{DC}$, $T_A = T_L$ to T_H unless otherwise noted)
See Figures 7-6 and 7-7.

Num	Characteristic	4 MHz		8 MHz		Unit
		Min	Max	Min	Max	
4 (3)	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (Setup)	50	—	50	—	ns
5	$\overline{CLK}$ Low to $\overline{DTACK}$ Low	—	220	—	90	ns
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ High	—	60	—	50	ns
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state	—	100	—	100	ns
9	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to Data Tri-state	—	50	—	50	ns
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (Hold Time)	0	—	0	—	ns
14	$\overline{IEI}$ Low to Falling CLK (Setup)	50	—	50	—	ns
15 (1)	$\overline{IEO}$ Valid from Clock Low (Delay)	—	180	—	120	ns
16	Data Valid from Clock Low (Delay)	—	300	—	180	ns
17	$\overline{IEO}$ Invalid from $\overline{IACK}$ High (Delay)	—	150	—	100	ns
18	$\overline{DTACK}$ Low from Clock High (Delay)	—	180	—	100	ns
19 (1)	$\overline{IEO}$ Valid from $\overline{IEI}$ Low (Delay)	—	100	—	100	ns
20	Data Valid from $\overline{IEI}$ Low (Delay)	—	220	—	140	ns
21	Clock Cycle Time	250	1000	125	1000	ns
22	Clock Width Low	110	—	55	—	ns
23	Clock Width High	110	—	55	—	ns
24 (4)	$\overline{CS}$, $\overline{IACK}$ Inactive to Rising Clock (Setup)	100	—	50	—	ns
26	$\overline{IACK}$ Width High/Minimum Delay between two Pulses	2	—	2	—	CLK

Notes:

1. $\overline{IEO}$ only goes low if no acknowledgeable interrupt is pending. If $\overline{IEO}$ goes low, $\overline{DTACK}$ and the data bus remain tri-stated.
2. If the setup time is not met, $\overline{CS}$ or $\overline{IACK}$ will not be recognized until the next falling CLK.
3. If the setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.

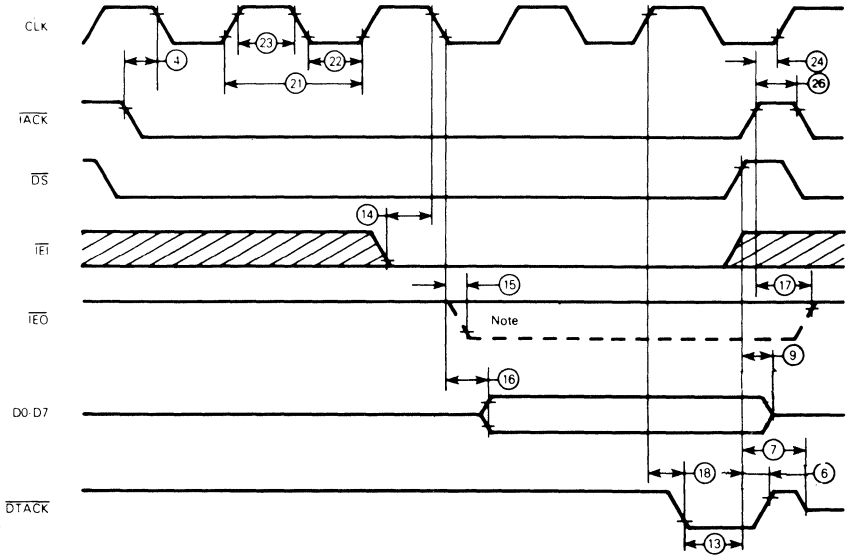
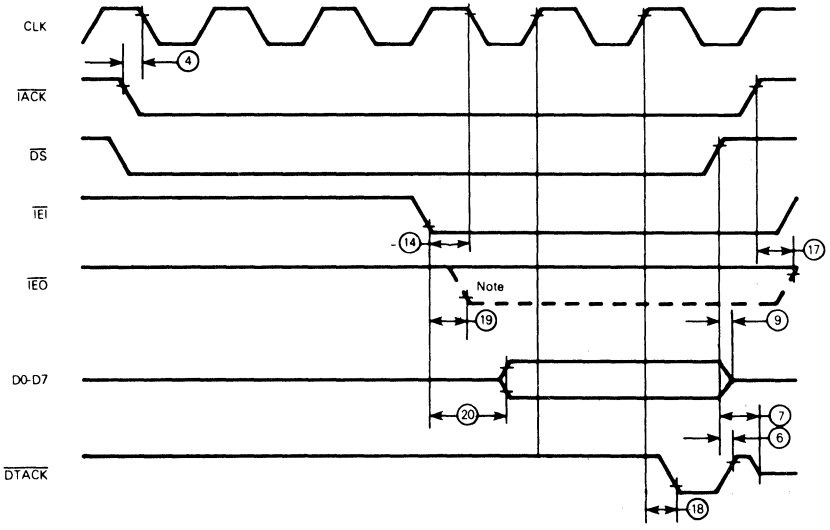


Figure 7-6. Interrupt Acknowledge Cycle ($\overline{IEI}$ Low)



Note: $\overline{CS}$ and $\overline{IACK}$ must be a function of $\overline{DS}$

Figure 7-7. Interrupt Acknowledge Cycle (IEI High)

7.7.4 AC ELECTRICAL CHARACTERISTICS - 6800 INTERFACE TIMING (V_{CC} = 5.0 V_{DC} ± 5%, V_{SS} = 0V_{DC}, T_A = 0°C to 70°C unless otherwise noted). See figure 7-8.

Num	Characteristic	4 MHz		8 MHz		Unit
		Min	Max	Min	Max	
51	Cycle Time	1000	—	—	—	ns
52	Pulse Width, E High	430	—	—	—	ns
53	Pulse Width, E Low	450	—	—	—	ns
54	Address, R/$\overline{W}$ Setup Time Before E	80	—	—	—	ns
55	$\overline{CS}$ Setup Time Before E	80	—	—	—	ns
56	Address Hold Time	10	—	—	—	ns
57	$\overline{CS}$ Hold Time	10	—	—	—	ns
58	Output Data Delay Time (Read)	—	250	—	—	ns
59	Data Hold Time (Read)	0	100	—	—	ns
60	Input Data Setup Time (Write)	280	—	—	—	ns
61	Data Hold Time (Write)	20	—	—	—	ns

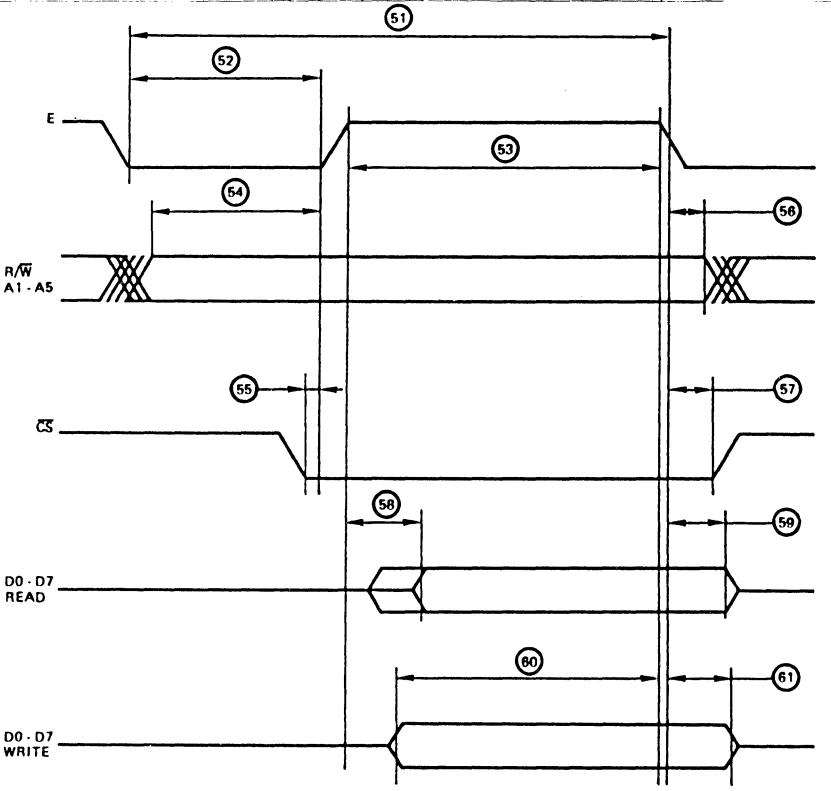


Figure 7-8. 6800 Interfacing Timing

7.7.5 AC ELECTRICAL CHARACTERISTICS - MULTIPLEXED BUS TIMING (V_{CC} = 5.0 V ± 5%, V_{SS} = 0 V_{DC}, T_A = 0°C to 70°C unless otherwise noted). See figures 7-9, 7-10

Num	Characteristic	4 MHz		8 MHz		Unit
		Min	Max	Min	Max	
62	Cycle Time	800	—	—	—	ns
63	Pulse Width $\overline{DS}$ Low or RD/WR High	350	—	—	—	ns
64	Pulse Width DS High or RD/WR Low	340	—	—	—	ns
65	Pulse Width AS/ALE High	100	—	—	—	ns
66	Delay AS Fall to $\overline{DS}$ Rise or ALE Fall to RD/WR Fall	30	—	—	—	ns
67	Delay $\overline{DS}$ or RD/WR Rise to AS/ALE Rise	30	—	—	—	ns
68	R/$\overline{W}$ Setup Time to $\overline{DS}$	100	—	—	—	ns
69	R/$\overline{W}$ Hold Time to $\overline{DS}$	10	—	—	—	ns
70	Address Setup Time to AS/ALE	20	—	—	—	ns
71	Address Hold Time to AS/ALE	20	—	—	—	ns
72	Data Setup Time to $\overline{DS}$ or WR (Write)	280	—	—	—	ns
73	Delay Data to $\overline{DS}$ or RD (Read)	—	250	—	—	ns
74	Data Hold Time to $\overline{DS}$ or WR (Write)	20	—	—	—	ns
75	Data Hold Time to $\overline{DS}$ or RD (Read)	0	100	—	—	ns
76	CE Setup Time to AS/ALE Fall	20	—	—	—	ns
77	CE Hold Time to $\overline{DS}$, RD or WR	20	—	—	—	ns

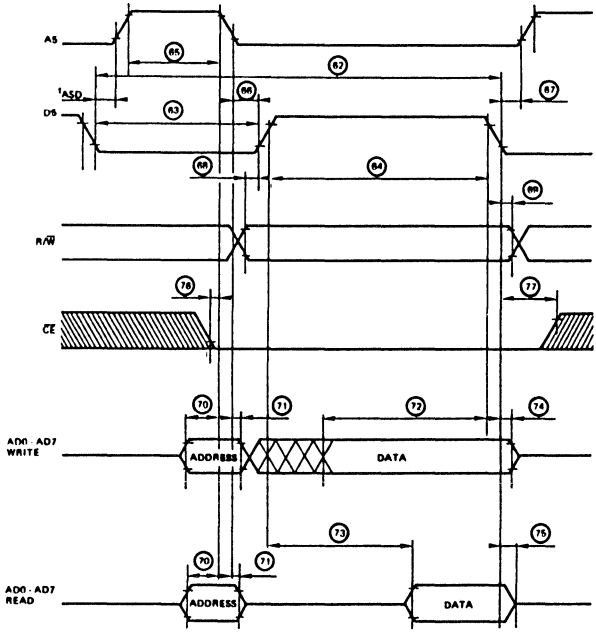


Figure 7-9. Multiplexed Bus Timi
Motorola Type

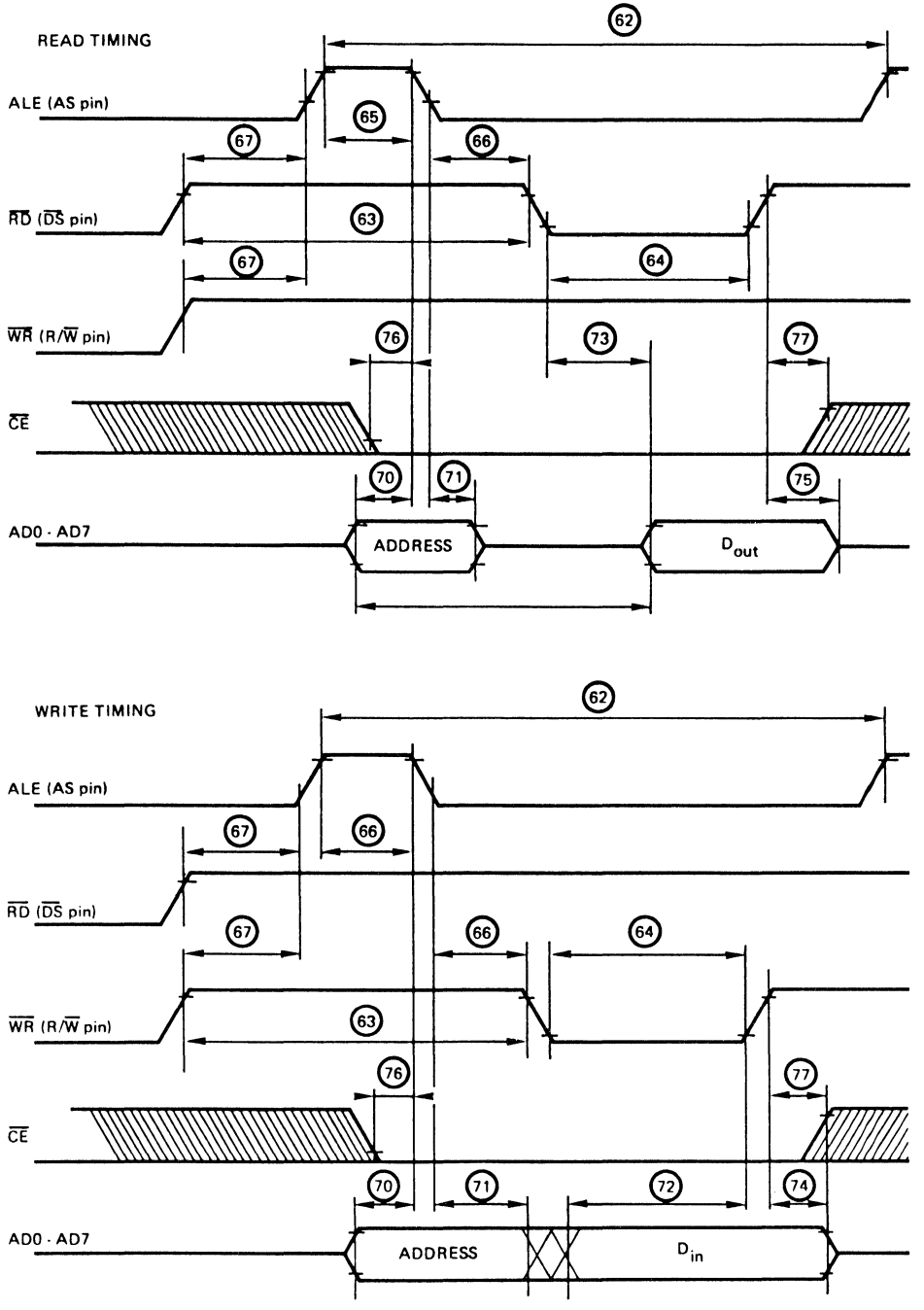
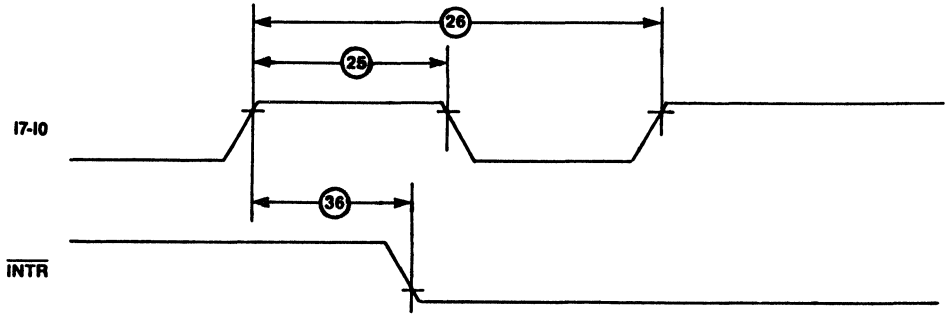


Figure 7-10. Multiplexed Bus Timing - Intel Type



Note: Active edge is assumed to be the rising edge

Figure 7-11. Interrupt Timing

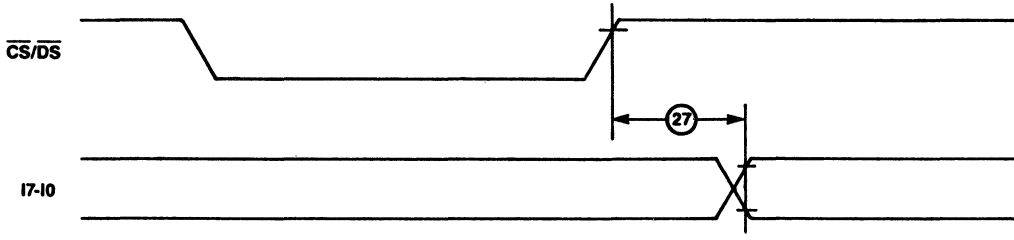


Figure 7-12. Port Timing

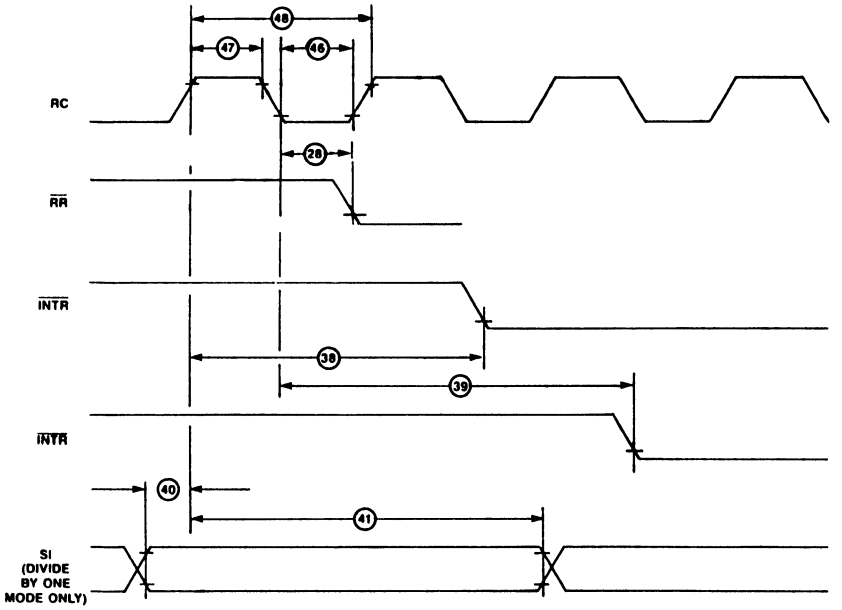


Figure 7-13. Receiver Timing

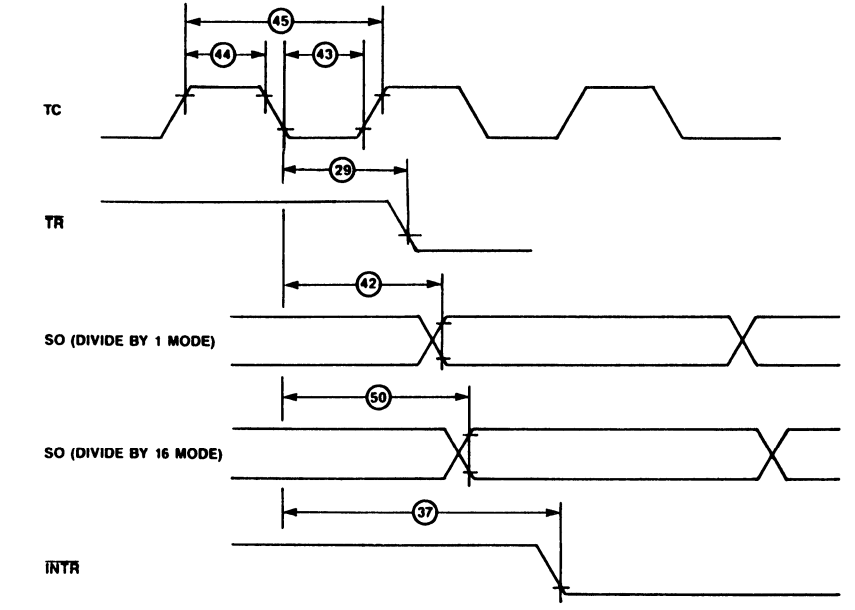


Figure 7-14. Transmitter Timing

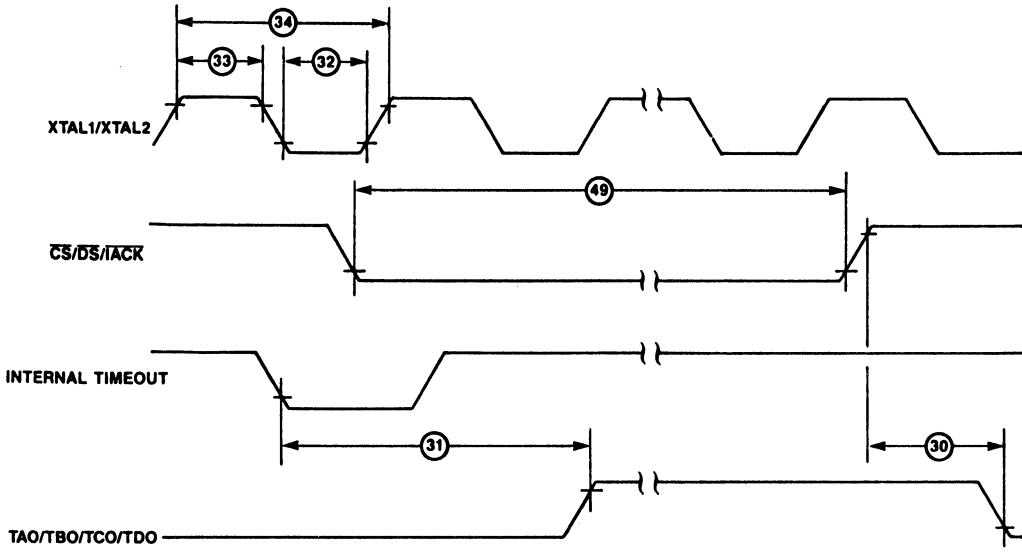


Figure 7-15. Timer Timing

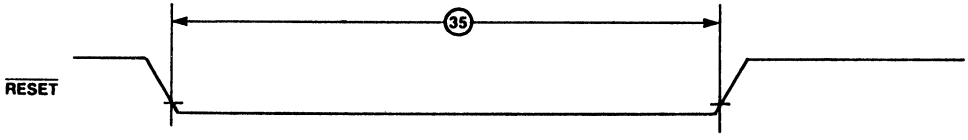


Figure 7-16. Reset Timing

7.8 TIMER AC CHARACTERISTICS

Definitions:

Error = Indicated time value − actual time value

$t_{psc} = t_{CLK} \times$ Prescale Value

Internal Timer Mode:

Single Interval Error (Free Running) (See Note 2)	± 100 ns
Cumulative Internal Error	0
Error Between Two Timer Reads	$\pm (t_{psc} - 4\ t_{CLK})$
Start Timer to Stop Timer Error	$2\ t_{CLK} + 100$ ns to $-(t_{psc} + 6\ t_{CLK} + 100$ ns)
Start Timer to Read Timer Error	0 to $-(t_{psc} + 6\ t_{CLK} + 400$ ns)
Start Timer to Interrupt Request Error (See Note 3)	$-2\ t_{CLK}$ to $-(4\ t_{CLK} + 800$ ns)

Pulse Width Measurement Mode:

Measurement Accuracy (See Note 1)	$2\ t_{CLK}$ to $-(t_{psc} + 4\ t_{CLK})$
Minimum Pulse Width	$4\ t_{CLK}$

Event Counter Mode:

Minimum Active Time of TAI and TBI	$4\ t_{CLK}$
Minimum Inactive Time of TAI and TBI	$4\ t_{CLK}$

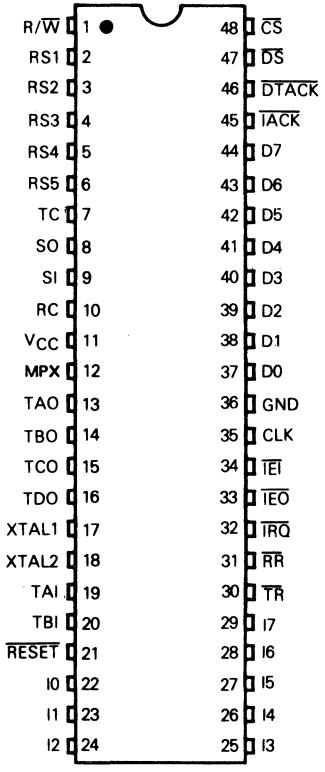
NOTES:

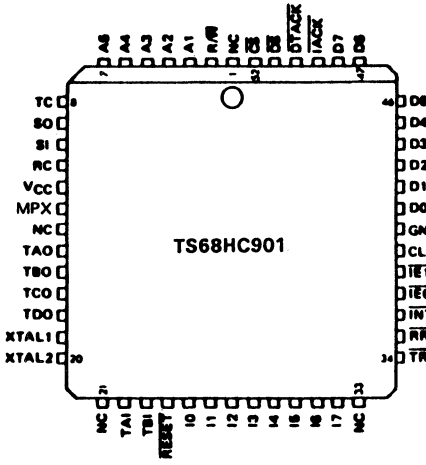
1. Error may be cumulative if repetitively performed.
2. Error with respect to t_{out} or $\overline{IRQ}$ if note 3 is true.
3. Assuming it is possible for the timer to make an interrupt request immediately.

SECTION 8
MECHANICAL DATA AND ORDERING INFORMATION

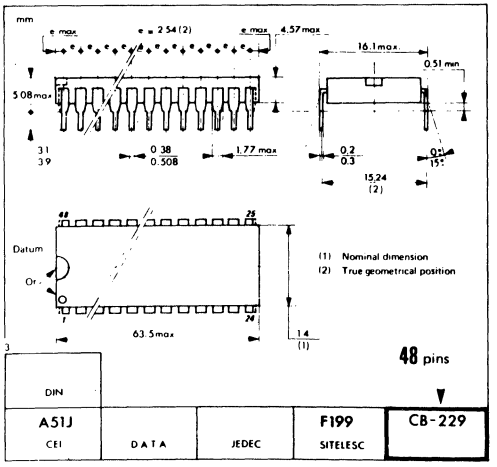
This section contains the pin assignments, package dimensions, and ordering information fo
the TS68HC901.

PIN ASSIGNMENTS

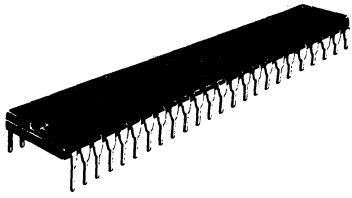




PHYSICAL DIMENSIONS

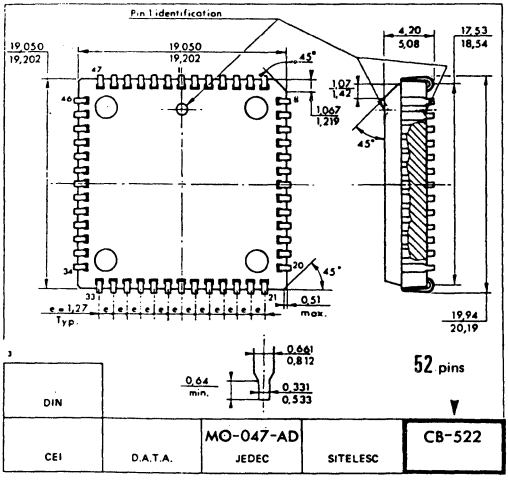


CB-229

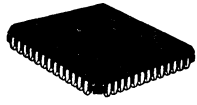


P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE
C SUFFIX
CERAMIC PACKAGE



CB-522



FN SUFFIX
PLCC 52

ORDERING INFORMATION

STANDARD VERSIONS

Package Type	Frequency (MHz)	Temperature Range	Part Number
Ceramic DIL	4.0	0°C to + 70°C	TS68HC901CC4
	5.0	0°C to + 70°C	TS68HC901CC5
C Suffix	8.0	0°C to + 70°C	TS68HC901CC8
Plastic DIL	4.0	0°C to + 70°C	TS68HC901CP4
	5.0	0°C to + 70°C	TS68HC901CP5
P Suffix	8.0	0°C to + 70°C	TS68HC901CP8
PLCC	4.0	0°C to + 70°C	TS68HC901CFN4
	5.0	0°C to + 70°C	TS68HC901CFN5
FN Suffix	8.0	0°C to + 70°C	TS68HC901CFN8

Hi-REL VERSIONS

In order to fit more closely to customer specific requirements, THOMSON SEMICONDUCTEURS is proposing different screening levels for its Hi-REL ranges.

G/B screening : Available only from THOMSON SEMICONDUCTEURS, this quality level, very close to the MIL-STD-883, is a cost effective alternative for customers who want to buy Hi-REL devices (low guaranteed AQL). The G/B level is in full accordance with the NFC96883 class G.

B/B screening : Full accordance with the MIL-STD-883 Rev. C, class B (US), the CECC 90,000 class B (european) and with the NFC96883 class B (French).

Details on screening procedures for these levels of selection are available on request (please contact our sales representatives).

Package Type	Frequency (MHz)	Temperature Range	Part Number
Ceramic DIL	4.0	— 40°C to + 85°C	TS68HC901VC4
C Suffix	4.0	— 55°C to + 125°C	TS68HC901MC4
	4.0	— 40°C to + 85°C	TS68HC901VCG/B4
	4.0	— 40°C to + 85°C	TS68HC901VCB/B4
	4.0	— 55°C to + 125°C	TS68HC901MCG/B4
	4.0	— 55°C to + 125°C	TS68HC901MCB/B4

CHAPTER 5 - MICROCOMPUTER PERIPHERALS

MICROCOMPUTER PERIPHERALS SELECTION GUIDE

Part number	Description	Technology	Alt source	CLK freq. (MHz)	Page
MK3801-0 MK3801-4 MK3801-6	Serial Timer Interrupt (STI) Controller - Full duplex USART - Two binary delay timers - Two full feature timers - Eight general purpose lines - Full control of each interrupt channel	NMOS	— — —	2.5 4 6	5-3
MK3831 MK3835	MCU - Real time clock - Serial I/O - 24 × 8 RAM - High speed shift clock - Low power CMOS - TTL compatible	CMOS	— —	4.19	5-19

THOMSON COMPONENTS MOSTEK

MK3801

SERIAL TIMER INTERRUPT CONTROLLER

FEATURES

- Full duplex USART with programmable DMA control signals
- Two binary delay timers
- Two full feature timers with
  - Delay to interrupt mode
  - Pulse width measurement mode
  - Event counter mode
- Eight general purpose lines with
  - Full bi-directional I/O capability
  - Edge triggered interrupts on either edge
- Full control of each interrupt channel
  - Enable/disable
  - Maskable
  - Automatic end-of-interrupt mode
  - Software end-of-interrupt mode
- 2.5, 4 MHz, and 6 MHz versions available

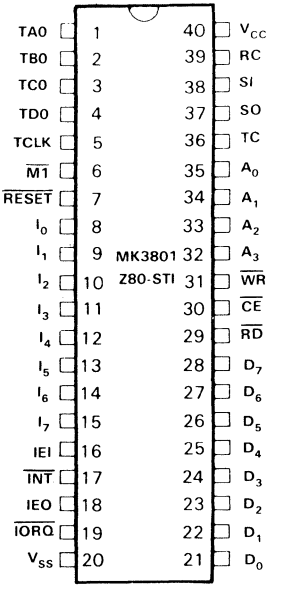
INTRODUCTION

The MK3801 Z80 STI (Serial Timer Interrupt) is a multifunctional peripheral device for use in Z80 microprocessor based systems. It is designed to optimize current systems by reducing chip count and system costs. By providing a USART, four timers (two binary and two full function), and eight bi-directional I/O lines with individually programmable interrupts, the MK3801 can make substantial improvement to any Z80 based system.

Control and operation of the MK3801 are provided by 24 internal registers accessible by the Z80 bus. Sixteen of these registers are directly addressable and accessible; eight are indirectly addressable. Two of the four timers provide full service features, while the other two provide delay timer features only. Serial Communication is provided by the USART, which is capable of either asynchronous or synchronous operation, optional sync word recognition and stripping, and programmable DMA control handshake lines. Eight bi-directional I/O lines provide parallel I/O capability and individually programmable interrupt capability. The interrupt structure of the device is fully programmable for all interrupts, provides for interrupt vector generation, conforms to the Z80 daisy chain interrupt priority scheme, and supports automatic end of interrupt functions for the Z80.

DEVICE PINOUT

Figure 1



SIGNAL NAME	DESCRIPTION
V_{SS}	Ground
V_{CC}	+5 volts (± 5 percent)
$\overline{CE}$	Chip Enable (Input, active low)
$\overline{RD}$	Read Enable (Input, active low)
$\overline{WR}$	Write Enable (Input, active low)
A_0-A_3	Address Inputs. Used to address one of the internal registers during a read or write operation
D_0-D_7	Data Bus (bi-directional)
$\overline{RESET}$	Device Reset (Input, active low). When activated, all internal registers (except for Timer or USART Data registers) will be cleared, all timers stopped, USART turned off, all interruptsdisabled and all pending interrupts cleared, and all I/O lines placed in tri-state input mode.
I_0-I_7	General purpose I/O and interrupt lines
$\overline{INT}$	Interrupt Request (Output, active low, open drain)
$\overline{IORQ}$	Input/Output Request from Z80-CPU (input, active low). The IORQ signal is used in conjunction with M1 to signal the MK3801 that the CPU is acknowledging its interrupt.
IEI	Interrupt Enable In, active High
IEO	Interrupt Enable Out, active High
SO	Serial Output
SI	Serial Input
RC	Receiver Clock Input
TC	Transmit Clock Input
TAO-TDO	Timer Outputs
TCLK	Timer Clock Input
$\overline{M1}$	Z80 Machine Cycle One (Input, active low)

PIN DESCRIPTION

Figure 1 illustrates the pinout of the MK3801. The functions of these individual pins are described above.

INTERNAL ORGANIZATION

Figure 2 illustrates the MK3801 internal organization, which supports the full set of timing, communications, parallel I/O, and interrupt processing functions available in the device.

CPU BUS I/O

The CPU BUS I/O provides the means of communications between the system and the MK3801. Data, Status, and Control Registers in the MK3801 are accessed by the bus in order to establish device parameters, assert control, and transfer status and data between the system and the MK3801.

Each register in the MK3801 is addressed over the address bus in conjunction with Chip Enable ($\overline{CE}$), while data is transferred over the eight bit Data bus under control of Read ($\overline{RD}$) and Write ($\overline{WR}$) signals.

REGISTER ACCESSES

All register accesses are independent of any system clock. To read a register, both $\overline{CE}$ and $\overline{RD}$ must be active. The internal read control signal is essentially the combination of both $\overline{CE}$ and $\overline{RD}$ active; thus the read operation will begin when the later of these two signals goes active and will end when the first signal goes inactive. The address bus must be stable prior to the start of the operation and must remain stable until the end of the operation. Unless a read operation or an interrupt acknowledge cycle is in progress, the data bus (D_0-D_7) will remain in the tri-state condition.

To write a register, both $\overline{CE}$ and $\overline{WR}$ must be active. The address must be stable prior to the start of the operation and must remain stable until the end of the operation. The data must be stable prior to the end of the operation and must remain stable until the end of the operation. The data presented on the bus will be latched into the register shortly after either $\overline{WR}$ or $\overline{CE}$ goes inactive.

INTERNAL REGISTERS

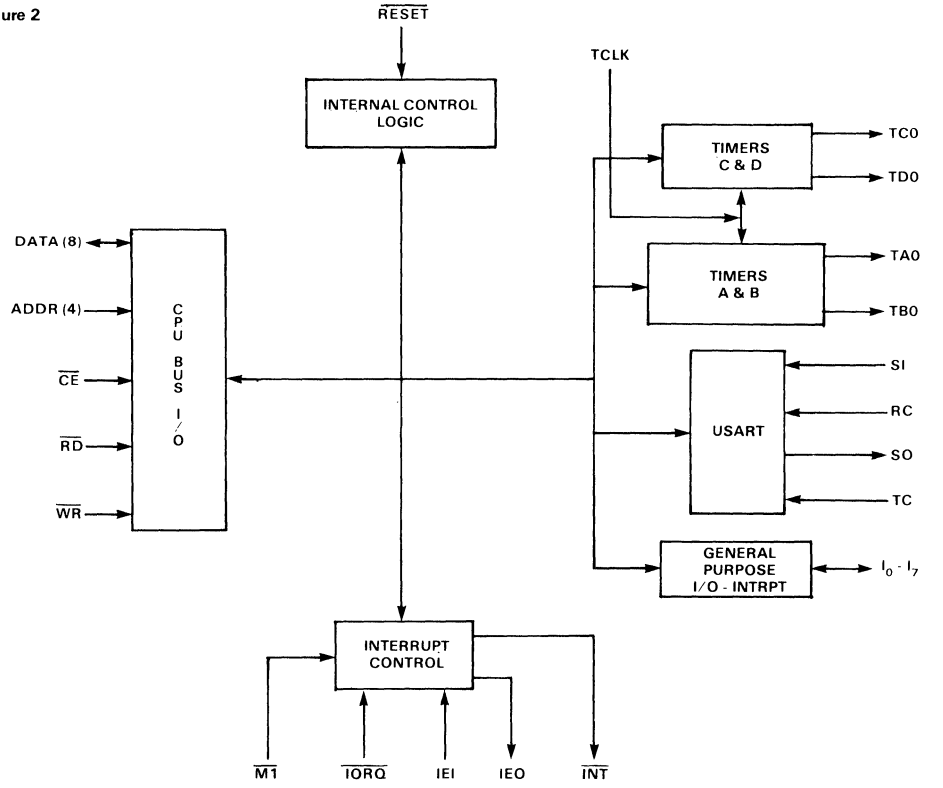
There are 24 internal registers used to control the operation of the STI. Sixteen of these registers are directly addressable and accessible. Eight registers are indirectly addressable via the Pointer/Vector Register and accessible via the Indirect Data Register.

DIRECTLY ADDRESSABLE REGISTERS

The Directly Addressable Registers are accessed by placing the address of the desired register on the address lines (A_0-A_3) during a write or read cycle. Figure 3 lists the Directly Addressable Registers.

INTERNAL ORGANIZATION

Figure 2



DIRECTLY ACCESSIBLE REGISTERS

Figure 3

ADDRESS	ABBREVIATION	REGISTER NAME
0	IDR	Indirect Data Register
1	GPIP	General Purpose I/O-Interrupt
2	IPRB	Interrupt Pending Register B
3	IPRA	Interrupt Pending Register A
4	ISRB	Interrupt in-Service Register B
5	ISRA	Interrupt in-Service Register A
6	IMRB	Interrupt Mask Register B
7	IMRA	Interrupt Mask Register A
8	PVR	Pointer/Vector Register
9	TABCR	Timers A and B Control Register

DIRECTLY ACCESSIBLE REGISTERS (Continued)
Figure 3

ADDRESS	ABBREVIATION	REGISTER NAME
A	TBDR	Timer B Data Register
B	TADR	Timer A Data Register
C	UCR	USART Control Register
D	RSR	Receiver Status Register
E	TSR	Transmitter Status Register
F	UDR	USART Data Register

INDIRECTLY ADDRESSABLE REGISTERS
Figure 4

INDIRECT ADDRESS	ABBREVIATION	REGISTER NAME
0	SCR	Sync Character Register
1	TDDR	Timer D Data Register
2	TCDR	Timer C Data Register
3	AER	Active Edge Register
4	IERB	Interrupt Enable Register B
5	IERA	Interrupt Enable Register A
6	DDR	Data Direction Register
7	TCDCR	Timers C and D Control Register

INDIRECTLY ADDRESSABLE REGISTERS

Indirectly Addressable Registers are addressed by placing the indirect address in bits IA0-IA2 of the Pointer/Vector Register, as defined in Figure 5. Data may be written to or read from the register indicated by these Indirect Register Address bits by a write or read access of the Indirect Data Register (selected when A_0-A_3 are all zero). The indirect address bits of the Pointer/Vector Register will remain unchanged after an indirect access. Repeated accesses of the Indirect Data Register will access the same indirect register as long as the indirect address in the Pointer/Vector Register remains unchanged. The Indirectly Addressable Registers are listed in Figure 4.

INTERRUPT VECTOR DEFINITION

Each individual function in the MK3801 is provided with a unique interrupt vector that is presented to the system during the interrupt acknowledge cycle. The interrupt vector returned during interrupt acknowledge is formed as shown in Figure 6. There are 16 vector addresses generated internally by the MK3801, one for each of the 16 interrupt channels. The three most significant bits of these vector addresses correspond to the three most significant bits of the Pointer/Vector Register shown in Figure 5. The least significant bit of each vector address is always 0, thus producing even vector addresses. The remaining 4 bits (IV_1 through IV_4) identify each of the 16 interrupt channels individually. The lowest priority channel responds with 0000 for IV_4-IV_1 respectively. The next higher priority channel responds with 0001, and so on in binary order, with the highest priority channel responding with 1111. Figure 7 lists each of the 16 interrupt channels in order of descending priority.

INTERRUPT CONTROL REGISTERS

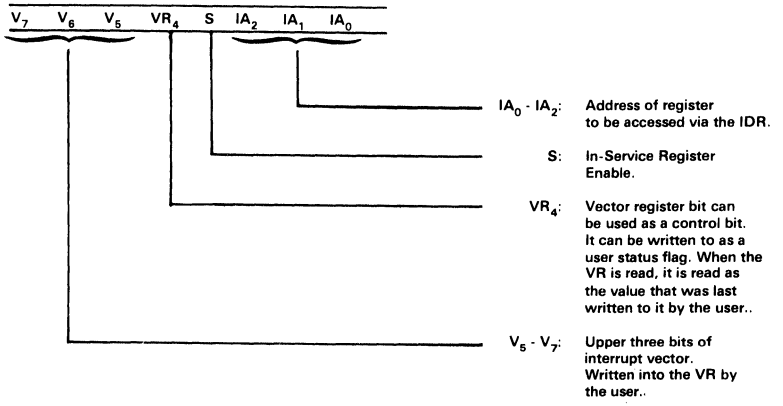
The Interrupt Control Registers provide control of interrupt processing for all I/O facilities of the MK3801. These registers allow the programmer to enable or disable any or all of the 16 interrupts, provide masking for any interrupts, and access to the pending or in-service status of the interrupts. Optional End-of-Interrupt modes are available under software control. The format of each of the Interrupt Control Registers is presented in Figure 8.

INTERRUPT OPERATION

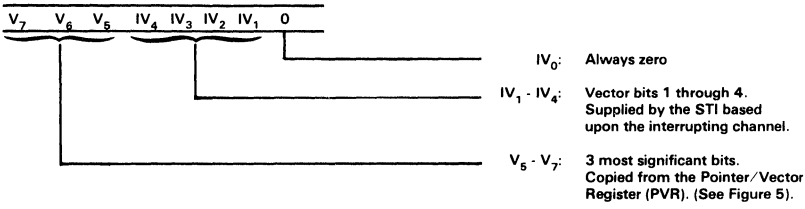
The Interrupt Enable Registers enable or disable the setting of an interrupt in the Interrupt Pending Registers. A 0 in a bit of the Interrupt Enable Registers disables the interrupt for the associated channel while a 1 enables the interrupt.

Once an interrupt is enabled, the occurrence of an interrupting condition on that channel will cause the

POINTER/VECTOR REGISTER (PVR) Port 08
Figure 5



INTERRUPT VECTOR
Figure 6



corresponding bit in the Interrupt Pending Register to be set. This indicates that an interrupt is pending in the MK3801.

Pending interrupts are presented to the Z80 CPU in order of priority (see Figure 1) unless they have been masked off. This is done by clearing the bit in the Interrupt Mask Register corresponding to the channel whose interrupt is to be masked. The channel's interrupt will remain pending until the mask bit for that channel is set, at which time the interrupt for that channel will be processsed in order of priority.

When an interrupt vector is generated for a pending interrupt and passed to the Z80 CPU, the bit in the Interrupt Pending Register, associated with the channel generating the interrupt, will be cleared. At this time, no history of the interrupt remains in the MK3801.

In order to retain historical evidence of an interrupt being serviced by the Z80, the In-Service Register may be enabled by setting the S-bit in the Pointer/Vector Register (see Figure 5). If the In-Service Register is enabled, the bit of the In-Service Register corresponding to the interrupting channel will be set when the interrupt vector is passed to the Z80. At the same time, the Interrupt Pending bit will be cleared since the interrupt is now in service. The In-Service bit will be cleared on execution of a Return-from-Interrupt (H'ED4D') instruction. The In-Service Registers are directly addressable, and the In-Service bit for any interrupt may be cleared by writing to the In-Service Register if the Return-from-Interrupt instruction is not used.

INTERRUPT CONTROL REGISTER DEFINITIONS

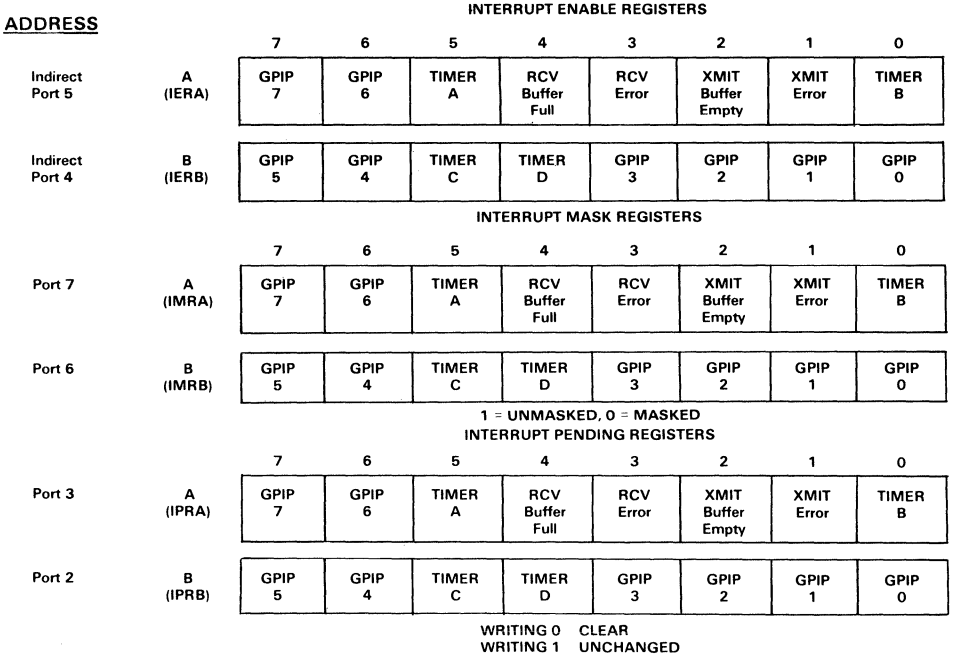
Figure 7

There are sixteen interrupt channels on the STI arranged in the following priority:

PRIORITY	CHANNEL	DESCRIPTION	ALTERNATE USAGE
HIGHEST	1111	General Purpose Interrupt 7 (I_7)	
	1110	General Purpose Interrupt 6 (I_6)	
	1101	Timer A	
	1100	Receive Buffer Full	
	1011	Receive Error	
	1010	Transmit Buffer Empty	
	1001	Transmit Error	
	1000	Timer B	
	0111	General Purpose Interrupt 5 (I_5)	
	0110	General Purpose Interrupt 4 (I_4)	TA (PW-Event)
	0101	Timer C	
	0100	Timer D	
	0011	General Purpose Interrupt 3 (I_3)	TB (PW-Event)
	0010	General Purpose Interrupt 2 (I_2)	
	0001	General Purpose Interrupt 1 (I_1)	DMA (TR)TX
LOWEST	0000	General Purpose Interrupt 0 (I_0)	DMA (RR)REC

INTERRUPT CONTROL REGISTERS

Figure 8



INTERRUPT CONTROL REGISTERS (Continued)

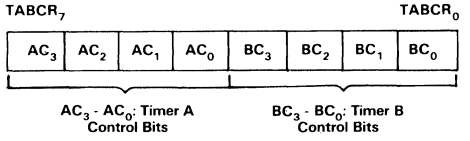
Figure 8

INTERRUPT SERVICE REGISTERS

ADDRESS		7	6	5	4	3	2	1	0
Port 5	A (ISRA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
Port 4	B (ISRB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

TIMER A and B CONTROL REGISTER (TABCR) Port 9

Figure 9

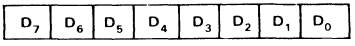


The four control bits are used to select the timer mode and prescale value, as follows:

CONTROL BIT DEFINITION

C_3	C_2	C_1	C_0	
0	0	0	0	Timer Stopped
0	0	0	1	Delay Mode, ÷4 Prescale
0	0	1	0	Delay Mode, ÷10 Prescale
0	0	1	1	Delay Mode, ÷16 Prescale
0	1	0	0	Delay Mode, ÷50 Prescale
0	1	0	1	Delay Mode, ÷64 Prescale
0	1	1	0	Delay Mode, ÷100 Prescale
0	1	1	1	Delay Mode, ÷200 Prescale
1	0	0	0	Event Count Mode
1	0	0	1	Pulse Width Mode, ÷4 Prescale
1	0	1	0	Pulse Width Mode, ÷10 Prescale
1	0	1	1	Pulse Width Mode, ÷16 Prescale
1	1	0	0	Pulse Width Mode, ÷50 Prescale
1	1	0	1	Pulse Width Mode, ÷64 Prescale
1	1	1	0	Pulse Width Mode, ÷100 Prescale
1	1	1	1	Pulse Width Mode, ÷200 Prescale

TIMER A DATA REGISTER AND TIMER B DATA REGISTER (TADR, TBDR) Port B & Port A



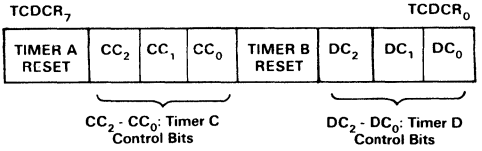
TIMERS

Four timers are available on the MK3801. Two provide full service features including delay timer operation, event counter operation, pulse width measurement operation, and pulse generation. The two other timers provide delay timer features only, and may be used for baud rate generators for use with the USART.

All timers are prescaler/counter timers, with a common independent clock input, and are not required to be operated from the system clock. In addition, all timers have a time-out output function that toggles each time the timer times out.

TIMER C and D CONTROL REGISTER (TCDCR) Indirect Port 7

Figure 10

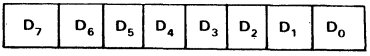


Three control bits are used to control each timer, as defined below:

CONTROL BIT DEFINITION

C_2	C_1	C_0	
0	0	0	Timer Stopped
0	0	1	Delay Mode, ÷4 Prescale
0	1	0	Delay Mode, ÷10 Prescale
0	1	1	Delay Mode, ÷16 Prescale
1	0	0	Delay Mode, ÷50 Prescale
1	0	1	Delay Mode, ÷64 Prescale
1	1	0	Delay Mode, ÷100 Prescale
1	1	1	Delay Mode, ÷200 Prescale

TIMER C DATA REGISTER and TIMER D DATA REGISTER (TCDR, TDDR) Indirect, Port 2 and Indirect Port 1



TIMER CONTROL REGISTERS

The 4 timers (A,B,C, and D) are programmed via 2 control registers and 4 timer data registers. Timers A and B are controlled by a single register (TABCR) and two timer data registers (TADR,TBDR). Timers C and D are controlled by a second control register (TCDCR) and two timer data

registers (TCDR, TDDR). Bits in the control registers allow the selection of operational mode, prescale, and control, while the data registers are used to read the timer or write the time constant register. General Purpose I/O Interrupt pins 3 (TB) and 4 (TA) are used for timer B and A inputs in event and pulse width modes. Figure 9 illustrates the Control and Data Register for timers A and B, while Figure 10 illustrates the Control and Data registers for timers C and D.

USART

Serial Communication is provided by the USART, which is capable of either asynchronous or synchronous operation. Variable word width and start/stop bit configurations are available under software control for asynchronous operation. For synchronous operation, a Sync Word is provided to establish synchronization during receive operations. The Sync Word will also be repeatedly transmitted when no other data is available for transmission. Operational modes exist to allow stripping of all Sync Words received in synchronous operation, and to allow the operation of DMA control handshake lines by the USART through General Purpose I/O Port lines 0 and 1. Separate receive and transmit clocks are available, and separate receive and transmit status and data bytes allow independent operation of the transmit and receive sections.

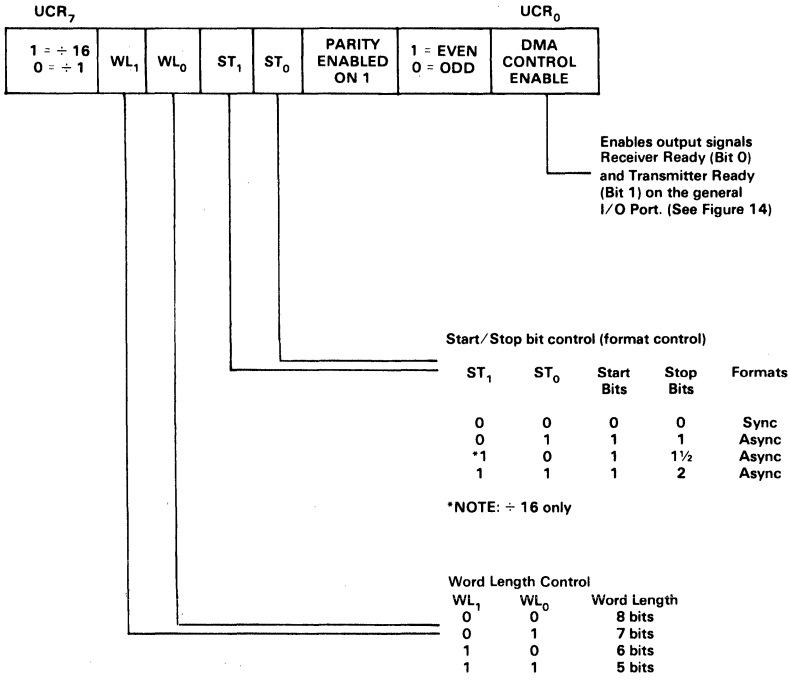
USART CONTROL REGISTERS

The USART is provided with 3 control/status registers and a data register. The programmer may specify operational parameters for the USART via the Control Register, as shown in Figure 11. Status of both the Receiver and Transmitter sections is accessed by means of the 2 Status Registers, as shown in Figure 12. Data written to the Data Register is passed to the transmitter, while reading the data register will access data received by the USART. The USART Data Register form is illustrated in Figure 13.

ERROR CONDITIONS

Error conditions in the USART are determined by monitoring the Receive Status Register (Port D) and the Transmitter Status Register (Port E). These error conditions are only valid for each word boundary and are not latched. When executing block transfers of data, it is necessary to save any errors so that they can be checked at the end of a block. In order to save error conditions during data transfer, the STI interrupt controller may be used by enabling error

USART CONTROL REGISTER (UCR) Port C
Figure 11



RECEIVER STATUS REGISTER (RSR) Port D

Figure 12

RSR_7							RSR_0
BUFFER FULL	OVERRUN ERROR	PARITY ERROR	FRAME ERROR	FOUND/$\overline{\text{SEARCH}}$ OR BREAK DETECT	MATCH/CHARACTER IN PROGRESS	SYNC STRIP ENABLE	RECEIVER ENABLE

TRANSMITTER STATUS REGISTER (TSR) Port E

TSR_7							TSR_0
BUFFER EMPTY	UNDERRUN ERROR	AUTO TURNAROUND	END OF TRANSMISSION	BREAK	HIGH	LOW	TRANSMITTER ENABLE

H	L	Serial Output State
0	0	Hi-Z
0	1	Low ("0")
1	0	High
1	1	Loop*

*Connects transmitter output to receiver input. In loopback mode, transmitter goes high when disabled. Also connects clocks with TC given priority.

USART DATA REGISTER (UDR) Port F

Figure 13

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

GENERAL PURPOSE I/O CONTROL REGISTERS

Figure 14

ACTIVE EDGE CONTROL REGISTER (AER) Indirect Port 3

1 = RISING
0 = FALLING

GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1	GPIP 0

DATA DIRECTION REGISTER (DDR) Indirect Port 6

1 = OUTPUT
0 = INPUT

GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1	GPIP 0

GENERAL PURPOSE I/O DATA REGISTER (GPIP) Port 1

GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1 (TR)	GPIP 0 (RR)
			TIMER A INPUT	TIMER B INPUT			

interrupts (Port 5, Indirect) for the desired channel (Receive error or Transmit error) and by masking these bits off (Port 7). Once the transfer is complete, the Interrupt Pending Register (Port 3) can be polled to determine the presence of a pending error interrupt, and therefore an error.

GENERAL PURPOSE I/O - INTERRUPT PORT

The General Purpose I/O - Interrupt Port provides eight I/O lines that may be operated either as inputs or outputs under software control. In addition, each line may generate an interrupt on either a positive going edge or a negative going edge of the input signal.

Two of the lines in this port provide auxiliary input functions for the timers in the pulse width measurement mode and the event counter mode. Two others serve as auxiliary output lines for the USART, one indicating the Receive Buffer Full condition (RR) and the other indicating the Transmitter Buffer Empty condition (TR). These may be used as handshake signals for a DMA controller or other external control circuitry.

GENERAL PURPOSE I/O CONTROL REGISTERS

The General Purpose I/O and Interrupt Port has 2 control registers. One allows the programmer to specify the Active Edge for each bit that will trigger the interrupt associated with that bit. The other register specifies the Data Direction (input and output) associated with each bit. The third register is the actual data I/O register used to input or output data to the port. When the USART is programmed to use DMA signals, this overrides the GPIP data and the DDR. The General Purpose I/O Control and Data Registers are illustrated in Figure 14.

MK3801 ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Temperature Under Bias ..... -25°C to + 100°C
Storage Temperature ..... -65°C to + 150°C
Voltage on Any Pin with Respect to Ground ..... -.3 V to + 7 V
Power Dissipation ..... 1.5 W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5 V ± 5% unless otherwise specified.

SYM	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
V_{IH}	Input High Voltage	2.0	V_{CC} + 3	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -120 μA
V_{OL}	Output Low voltage		0.4	V	I_{OL} = 2.0 mA
I_{LL}	Power Supply Current		180	mA	Outputs Open
I_{LI}	Input Leakage Current		±10	μA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float		10	μA	V_{OUT}=2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float		-10	μA	V_{OUT} = 0.4 V

All voltages are referenced to ground.

CAPACITANCE

T_A = 25°C, f = 1 MHz unmeasured pins returned to ground.

SYM	PARAMETER	MAX	UNIT	TEST CONDITION
C_{IN}	Input Capacitance	10	pf	Unmeasured pins returned to ground
C_{OUT}	Tri-state Output Capacitance	10	pf	

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5 V ± 5% unless otherwise noted.

SIGNAL	SYMBOL	PARAMETER	MK3801-0		MK3801-4		MK3801-6		UNIT	CONDITION
			MIN	MAX	MIN	MAX	MIN	MAX		
A_0-A_3	T_{SAR} & T_{SAW}	Address setup time prior to falling edge of $\overline{CEWR}$ or $\overline{CERD}$	80		30		15		ns	
	T_{HAR} & T_{HAW}	Address hold time after rising edge of $\overline{CEWR}$ or $\overline{CERD}$	0		0		0		ns	
$\overline{CEWR}$	T_{WL}	$\overline{CEWR}$ pulse width low (write cycle)	360		205		175		ns	Note 1
	T_{WW}	$\overline{CEWR}$ high time between write cycles	580		400		300		ns	
	T_{WRD}	$\overline{CEWR}$ high to $\overline{CERD}$ low	580		400		300		ns	
$\overline{CERD}$	T_{RDL}	$\overline{CERD}$ pulse width low (read cycle)	400		250		215		ns	Note 1
	T_{RR}	$\overline{CERD}$ high time between read cycles	300		200		190		ns	
	T_{M1RD}	Rising $\overline{M1RD}$ to falling $\overline{M1RD}$	225		165		95		ns	
	T_{RDW}	$\overline{CERD}$ high to $\overline{CEWR}$ low	125		100		75			
$\overline{M1}$	T_{SM1}	$\overline{M1}$ setup time prior to falling $\overline{IORQ}$ during interrupt acknowledge	800		500		350		ns	
$\overline{IORQ}$	T_{IOL}	$\overline{IORQ}$ low time	300		185		170		ns	
IEI	T_{SIEI}	Setup to falling $\overline{IORQ}$ during interrupt acknowledge	140		80		65		ns	
	T_{SRD}	Setup prior to end of 4D read on RETI	100		50		40		ns	
D_0-D_7	T_{SDM1}	Data valid prior to rising $\overline{RD}$ ($\overline{M1}$ cycle)	65		50		45		ns	Load 100 pf + 1 TTL load
	T_{HDM1}	Data hold time after rising $\overline{RD}$ ($\overline{M1}$ cycle)	0		0		0		ns	
	T_{DRD}	Data output delay from $\overline{CERD}$		400		250		215	ns	
	T_{SDW}	Data setup time to rising edge of $\overline{CEWR}$	350		280		175		ns	
	T_{HDW}	Data hold time from rising edge of $\overline{CEWR}$	0		0		0		ns	
	T_{DDI}	Data output delay from falling $\overline{IORQ}$ during interrupt acknowledge		300		185		170	ns	

A.C. CHARACTERISTICS (Continued)

SIGNAL	SYMBOL	PARAMETER	MK3801-0		MK3801-4		MK3801-6		UNIT	CONDITION
			MIN	MAX	MIN	MAX	MIN	MAX		
	T_{DHVZ}	Data hold time following M1 IORQ during interrupt acknowledge cycle.	0		0		0		ns	
	T_{DDZ}	Delay to float		150		100		80	ns	
I_0-I_7	T_{IPW}	Minimum active pulse width	200		100		90		ns	
	T_{ICY}	Minimum time between active edges	200		100		90		ns	
	T_{DIW}	Data valid from rising CEWR		600		500		400	ns	Load 100 pf + 1 TTL
RR	T_{DRR}	Delay from rising RC		710		590		545	ns	1 TTL
TR	T_{DTR}	Delay from rising TC		800		645		590	ns	
TAO-TDO	T_{DTW}	Timer output low from rising edge of CEWR (A & B) (Reset T_{OUT})		600		500		400	ns	Load 100 pf +
	T_{DTI}	T_{OUT} valid from Internal timeout		$2\ t_{CLK}$ +400		$2\ t_{CLK}$ +300		$2\ t_{CLK}$ +250	ns	1 TTL load
TCLK	$T_{t_{CLKL}}$	Low time	130		95		75		ns	
	$T_{t_{CLKH}}$	High time	130		95		75		ns	
	$T_{t_{CKC}}$	Cycle time	300	2500	200	2500	165	2500	ns	
RESET	T_{RSL}	Low time for part reset	3		2		1.6		μs	
IEO	T_{DIEOH}	IEO delay from rising edge of IEI		200		130		100	ns	Load 100 pf +
	T_{DIEOL}	IEO delay from falling edge of IEI		200		130		100	ns	1 TTL load
	T_{DIEOM}	IEO delay from falling edge of M1 (interrupt occurring just prior to M1)		270		190		110	ns	
	T_{DIEOA}	Delay to rising IEO from rising IORQ during interrupt acknowledge		1000		800		600	ns	
	T_{DIEOR}	Delay to rising IEO from rising edge of RD during ED fetch of RETI		500		400		300	ns	
INT	T_{DRCI}	Time delay from Receive Clock to interrupt from rising or falling edge of RC		910		760		680	ns	

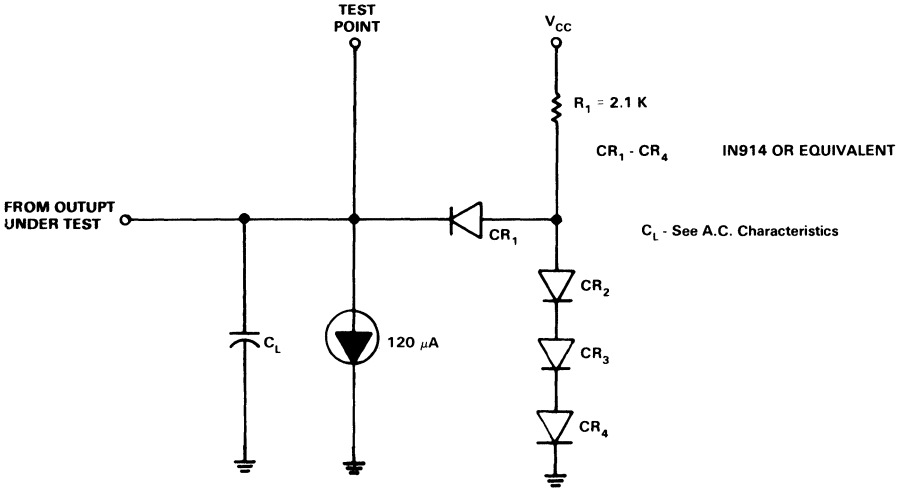
A.C. CHARACTERISTICS (Continued)

SIGNAL	SYMBOL	PARAMETER	MK3801-0		MK3801-4		MK3801-6		UNIT	CONDITION
			MIN	MAX	MIN	MAX	MIN	MAX		
	T_{DTCI}	Time delay from Transmit Clock to interrupt from rising or falling edge of TC		1460		980		690	ns	
SI	T_{SSI}	Serial in set up time to rising edge of RC (Divide by one only)	80		80		55		ns	
	T_{HSI}	Data hold time from rising edge of RC (Divide by one only)	400		350		300		ns	
SO ÷ 1	T_{DSO}	Data valid from falling edge of TC		420		390		345	ns	100 pf + 1 TTL load
SO ÷ 16	T_{DSO}	Data valid from falling edge of TC		520		490		445	ns	100 pf + 1 TTL load
TC	T_{TCL}	Low time	650		500		400		ns	
	T_{TCH}	High time	650		500		400		ns	
	T_{TCCY}	Cycle time	1.5		1.05		.85		μs	
RC	T_{RCL}	Low time	650		500		400		ns	
	T_{RCH}	High time	650		500		400		ns	
	T_{RCCY}	Cycle time	1.5		1.05		.85		μs	

NOTE:
1. One wait state must be inseted when used as a 6 MHz memory mapped device.
2. All A.C. measurements are referenced to V_{IL} max., V_{IH} min., V_{08} (0.8V), or (2.0 V).

OUTPUT LOAD CIRCUIT

Figure 15



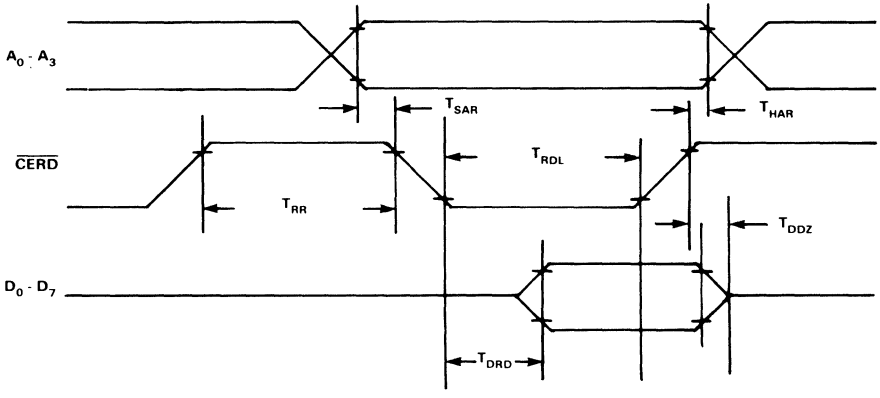
TIMING DIAGRAMS

Figure 16

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"		"0"
OUTPUT	2.0 V		0.8 V
INPUT	2.0 V		0.8 V
FLOAT	△ V	=	0.5 V

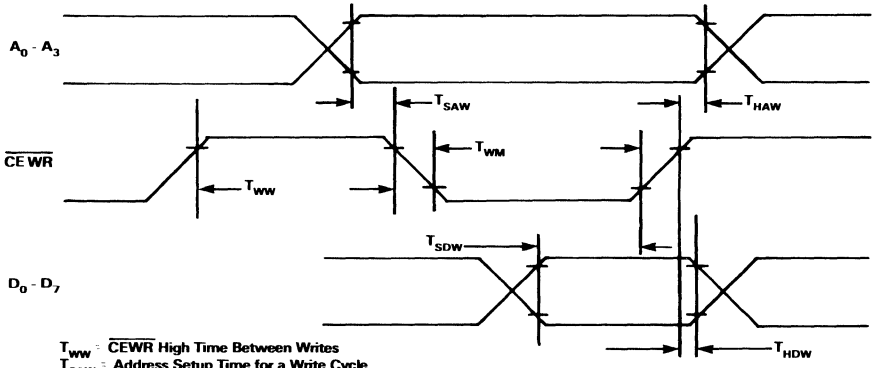
READ CYCLE



T_{SAR} - Address Setup Time for a Read Cycle
T_{DRD} Data Output Delay from $\overline{CERD}$
T_{DDZ} Time to Tri-State Following a Read Cycle
T_{HAR} Required Address Hold Time Following a Read Cycle

WRITE CYCLE

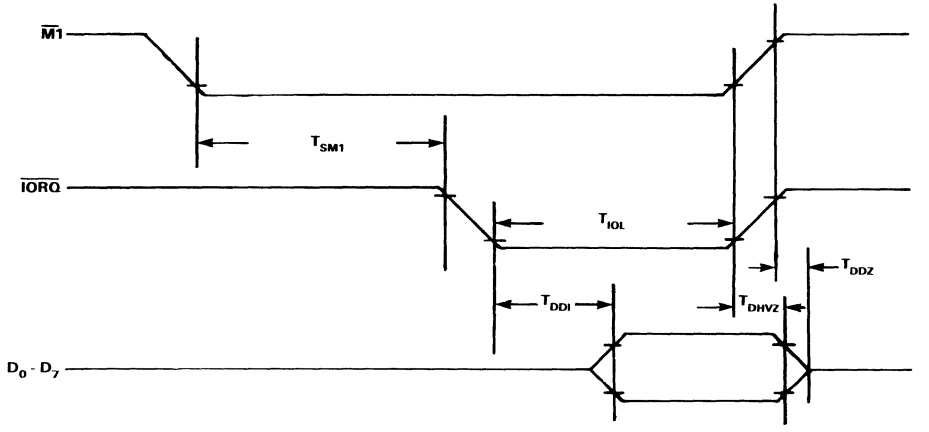
Figure 17



T_{WW} = $\overline{CEWR}$ High Time Between Writes
T_{SAW} = Address Setup Time for a Write Cycle
T_{SDW} = Data Setup Time Prior to the End of a Write Cycle
T_{HDW} = Required Data Hold Time Following a Write Cycle
T_{HAW} = Required Address Hold Time Following a Write Cycle
T_{WM} = $\overline{CEWR}$ pulse width low

INTERRUPT ACKNOWLEDGE CYCLE

Figure 18



T_{IOL} = $\overline{IORQ}$ Pulse Width Low
T_{SMI} = $\overline{M1}$ Setup Time prior to $\overline{IORQ}$ For an Acknowledge cycle
T_{DDI} = Access Time for Vector
T_{DDZ} = Time to Tri-State Following a Vector
T_{DHVZ} = Data hold time following $\overline{M1}$ $\overline{IORQ}$ during interrupt acknowledge cycle

TIMER A.C. CHARACTERISTICS

Definitions:

Error = Indicated Time Value - Actual Time Value

tpsc = t_{CLK} x Prescale Value

Internal Timer Mode

	V_{CC} ± 10%
Single Interval Error (free running) (Note 2)	± 100 ns
Cumulative Internal Error	0
Error Between Two Timer Reads	± (tpsc + 4 t_{CLK})
Start Timer to Stop Timer Error	2 t_{CLK} + 100 ns to -(tpsc + 6t_{CLK} + 100 ns)
Start Timer to Read Timer Error	0 to -(tpsc + 6 t_{CLK} + 400 ns)
Start Timer to Interrupt Request Error (Note 3)	-2 t_{CLK} to -(4t_{CLK} + 800 ns)

Pulse Width Measurement Mode

Measurement Accuracy (Note 1)	2 t_{CLK} to -(tpsc + 4t_{CLK})
Minimum Pulse Width	4t_{CLK}

Event Counter Mode

Minimum Active Time of I_3, I_4	4t_{CLK}
Minimum Inactive Time of I_3, I_4	4t_{CLK}

NOTES:
1. Error may be cumulative if repetitively performed.
2. Error with respect to T_{OUT} or INT if note 3 is true.
3. Assuming it is possible for the timer to make an interrupt request immediately.

ORDERING INFORMATION

PART NO.	DESIGNATOR	PACKAGE TYPE	MAX CLOCK FREQUENCY	TEMPERATURE RANGE
MK3801N-0	Z80-STI	Plastic	2.5 MHz	0 to 70°C
MK3801N-4	Z80-STI	Plastic	4.0 MHz	0 to 70°C
MK3801N-6	Z80-STI	Plastic	6.0 MHz	0 to 70°C

MK3835N • MK3831N

CMOS MICROCOMPUTER CLOCK/RAM

FEATURES

- Real-time clock counts seconds, minutes, hours, date of the month, day of the week, month, and year. Every 4th year, February has 29 days.
- Serial I/O for minimum pin count (8 pins)
- 24 x 8 RAM for scratchpad data storage
- Simple Microcomputer interface
- High speed shift clock independent of crystal oscillator frequency
- Single byte or multiple byte (Burst Mode) data transfer capability for read or write of clock or RAM data
- TTL Compatible (V_{CC} = 5V)
- Low-power CMOS
- MK3831 is available with fixed frequency operation (4.194304 MHz) and low power operation due to the disabling of CKO (Pin 1).

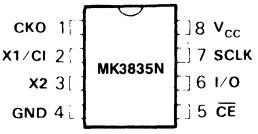


Figure 1. Pin Out

PIN DESCRIPTION

Table 1

PIN 3835N	NAME	DESCRIPTION
1	CKO	Buffered System Clock Output
2	X1/CI	Crystal or External Clock Input
3	X2	Crystal Input
4	GND	Power Supply Pin
5	$\overline{CE}$	Chip Enable for Serial I/O Transfer
6	I/O	Data Input/Output Pin
7	SCLK	Shift Clock for Serial I/O Transfer
8	V_{CC}	Power Supply Pin

GENERAL DESCRIPTION

Many microprocessor applications require a real-time clock and/or memory that can be battery powered with very low power drain. The MK3835N is specifically designed for these applications. The device contains a real-time clock/calendar, 24 bytes of static RAM, an on-chip oscillator, and it communicates with the microprocessor via a simple serial interface. The MK3835N is fabricated using CMOS technology, thus ensuring very low power consumption.

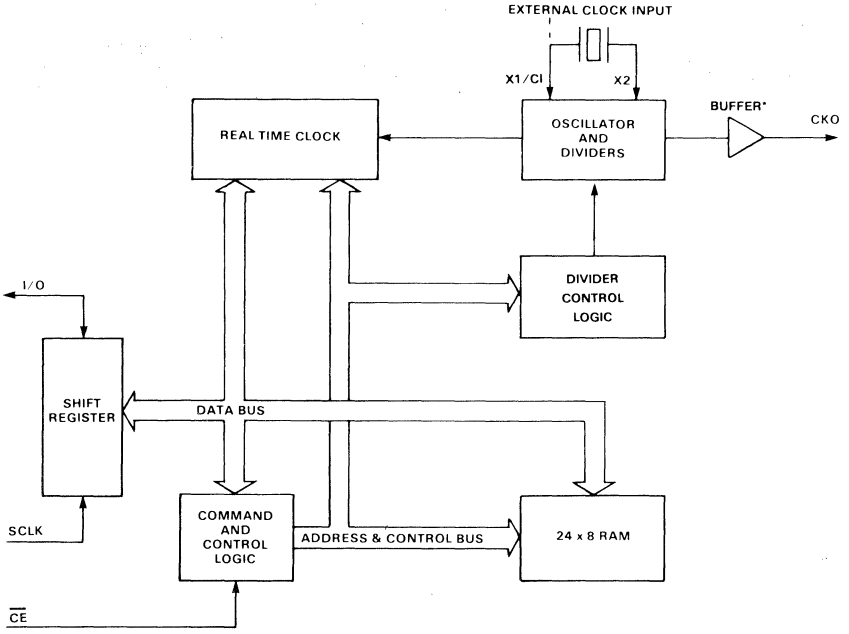
The real-time clock/calendar provides seconds, minutes, hours, day, date, month, and year information to the microprocessor. The end of the month date is automatically adjusted for months with less than 31 days, including correction for leap year every 4 years. The clock operates in either the 24 hour or 12 hour format with an AM/PM indicator.

The on-chip oscillator provides a real-time clock source for the clock/calendar. It incorporates a programmable divider so that a wide variety of crystal frequencies can be accommodated. The oscillator also has an output available that can be connected to the microprocessor clock input. A separately programmable divider provides several different output frequencies for any given crystal frequency. This feature can eliminate having to use a separate crystal or external oscillator for the microprocessor, thereby reducing system cost.

Interfacing the CLOCK/RAM with a microprocessor is greatly simplified using synchronous serial communication. Only 3 lines are required to communicate with the CLOCK/RAM: (1) $\overline{CE}$ (chip enable), (2) I/O (data line) and (3) SCLK (shift register clock). Data can be transferred to and from the CLOCK/RAM one byte at a time or in a burst of up to 24 bytes.

TECHNICAL DESCRIPTION

Figure 2 is a block diagram of the CLOCK/RAM chip. Its main elements are the oscillator and divider circuit, divider control logic, the real-time clock/calendar, static RAM, the serial shift register, and the command and control logic.



* BUFFER DISABLED IN THE MK3831

Figure 2. Block Diagram

The shift register is used to communicate with the outside world. Data on the I/O line is either input or output on each shift register clock pulse when the chip is enabled. If the chip is in the input mode, the data on the I/O line is input to the shift register on the rising edge of SCLK. If in the output mode, data is shifted out onto the I/O line on the falling edge of SCLK.

The command and control logic receives the first byte input by the shift register after $\overline{CE}$ goes active. This byte must be the command byte and will direct further operations within the CLOCK/RAM. The command specifies whether subsequent transfers will be data input or data output, and which register or RAM location will be involved.

A control register provides programmable control of the divider for the internal clock signal, the external clock signal, the crystal type and mode, and the write protect function.

The real-time clock/calendar is accessed via seven dynamic registers. These registers are seconds, minutes, hours, day, date, month, and year. Certain bits within these registers also control a run/stop function, 12/24 hour format, and indicate AM or PM (12 hour mode only). These registers can be accessed sequentially in Burst Mode, or randomly in a single byte transfer.

The static RAM is organized as 24 bytes of 8-bits each. They can be accessed either sequentially in burst mode, or randomly in a single byte transfer.

POWER UP

A time base on the crystal input pins is necessary for correct power up. This time base can be provided by a crystal or it can be derived from another generated clock source. It should be noted that a delay exists between power up and the correct power up state of the clock and control registers.

DATA TRANSFER

Data Transfer is accomplished under control of the $\overline{CE}$

and SCLK inputs by an external microcomputer. Each transfer consists of a single byte ADDRESS/COMMAND input followed by a single byte or multiple byte (if Burst Mode is specified) data input or output, as specified by the ADDRESS/COMMAND byte. The serial data transfer occurs with LSB first, MSB last format.

ADDRESS/COMMAND BYTE

The ADDRESS/COMMAND Byte is shown below:

7	6	5	4	3	2	1	0
1	RAM $\overline{CK}$	A4	A3	A2	A1	A0	Rd $\overline{W}$

As defined, the MSB (bit 7) must be a logical 1; bit 6 specifies a Clock/Calendar/Control register if logical 0 or a RAM register if logical 1; bits 1-5 specify the designated register(s) to be input or output; and the LSB (bit 0) specifies a WRITE operation (input) if logical 0 or READ operation (output) if logical 1.

BURST MODE

Burst Mode may be specified for either the Clock/Calendar/Control registers or for the RAM registers by addressing location 31 Decimal (ADDRESS/COMMAND bits 1-5 = logical 1). As before, bit 6 specifies Clock or RAM and bit 0 specifies read or write.

There is no data storage capability at location 31 in either the Clock/Calendar/Control registers or the RAM registers.

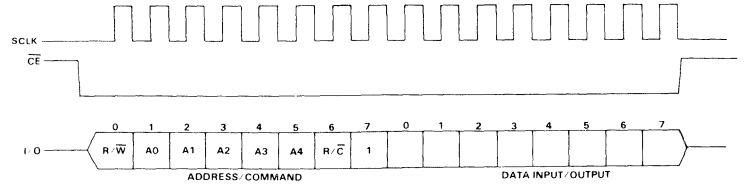
SCLK AND $\overline{CE}$ CONTROL

All data transfers are initiated by $\overline{CE}$ going low. After $\overline{CE}$ goes low, the next 8 SCLK cycles input an ADDRESS/COMMAND byte of the proper format. An SCLK cycle is the sequence of a positive edge followed by a negative edge. For data inputs, the data must be valid during the SCLK cycle. If bit 7 is not logical 1, indicating a valid CLOCK/RAM ADDRESS/COMMAND, the ADDRESS/COMMAND byte is ignored as are all SCLK cycles until $\overline{CE}$ goes high and returns low to initiate a new ADDRESS/COMMAND transfer. See Figure 3.

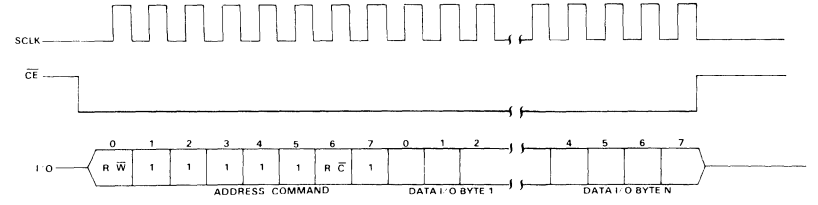
ADDRESS/COMMAND bits and DATA bits are input on the rising edge of SCLK, and DATA bits are output on the falling edge of SCLK.

A data transfer terminates if $\overline{CE}$ goes high, and the transfer must be reinitiated by the proper ADDRESS/COMMAND when $\overline{CE}$ again goes low. The data I/O pin is high impedance when $\overline{CE}$ is high.

I. SINGLE BYTE TRANSFER



II. BURST MODE TRANSFER



NOTES

1) Data input sampled while SCLK is high
2) Data output changes on falling edge of clock
3) Rising edge of $\overline{CE}$ terminates operation and resets address/command

FUNCTION	BYTE N	SCLK n
CLOCK	8	72
RAM	24	200

Figure 3. Data Transfer Summary

DATA INPUT

Following the 8 SCLK cycles that input the WRITE Mode ADDRESS/COMMAND byte (bit 0 = logical 0), a DATA byte is input on the rising edge of the next 8 SCLK cycles (per byte, if Burst Mode is specified). Additional SCLK cycles are ignored should they inadvertently occur.

DATA OUTPUT

Following the 8 SCLK cycles that input the READ Mode ADDRESS/COMMAND byte (bit 0 = logical 1), a DATA byte is output on the falling edge of the next 8 SCLK cycles (per byte, if Burst Mode is specified). Note that the first data bit to be transmitted from the CLOCK/RAM occurs on the falling edge of the last bit of the command byte. Additional SCLK cycles retransmit the data byte(s) should they inadvertently occur, so long as $\overline{CE}$ remains low. This operation permits continuous Burst Read Mode capability.

DATA TRANSFER SUMMARY

A data transfer summary is shown in Figure 3.

REGISTER DEFINITION

CLOCK/CALENDAR

The Clock/Calendar is contained in 7 writeable/readable registers, as defined below.

Address	Function	Range (BCD)
0	Seconds+Clock Halt Flag	00-59
1	Minutes	00-59
2	Hours/AM-PM/12-24 Mode	00-23 or 01-12
3	Date	01-28,29, 30,31
4	Month	01-12
5	Day	01-07
6	Year	00-99

Data contained in the Clock/Calendar registers is in binary coded decimal format (BDC).

CLOCK HALT FLAG

Bit 7 of the Seconds Register is defined as the Clock Halt Flag. Bit 7 = logical 1 inhibits the 1 Hz input to the Clock/Calendar. Bit 7 is set to logical 1 on power-up to prevent counting, and it may be set high or low by writing to the seconds register under normal operation of the device.

AM-PM/12-24 MODE

Bit 7 of the Hours Register is defined as the 12 or 24 hour mode select bit. When high, the 12 hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20-23 hours).

TEST MODE BITS

Bit 7 of the Date Register and Bit 7 of the Day Register are Test Mode Bits utilized in testing the MK3835N. These bits should be logic 0 for normal operation.

CONTROL REGISTER

The Control Register specifies the crystal mode/frequency to be used, the system clock output frequency, and the WRITE PROTECT Mode for data protection. The Control Register is located at address 7 in the Clock/Calendar/Control address space.

7	6	5	4	3	2	1	0
WP	C1	C0	X4	X3	X2	X1	X0

CRYSTAL DIVIDER MODE

X4 and X3 specify the Crystal frequency divider mode selected.

X4	X3	Xtal Mode	Primary Frequencies
0	0	Binary	2^{22}, 2^{21}, 2^{20} Hz
0	1	Microprocessor	8, 5, 4, 2.5, 2, 1.25, 1 MHz
1	0	Baud Rate	7.3728, 3.6864, 1.8432 MHz
1	1	Color Burst	3.5795 MHz

CRYSTAL DIVIDER PRESCALER

X2, X1, and X0 specify a particular prescaler divider selection necessary to generate a 1 Hz frequency for the Clock/Calendar. Refer to Table 2 for complete definition.

SYSTEM CLOCK OUTPUT

C1 and C0 designate the system clock output frequency selected. The options are X, X/2, X/4, and ~2 kHz. When in the Binary Mode and C1, C0 = '1', the output frequency is 2048 Hz. In any other mode the output frequency is ~2048 Hz. Refer to Table 3 for complete definition.

WRITE PROTECT

Bit 7 of the Control Register is the WRITE PROTECT

Flag. Bit 7 is set to logical 1 on power-up, and it may be set high or low by writing to the Control Register. When high, the WRITE PROTECT Flag prevents a write operation to any internal register, including the other bits of the Control Register. Further, logic is included such that the WRITE PROTECT bit may be reset to a logic 0 by a Write operation without altering the other bits of the Control Register.

MK3831 CONTROL REGISTER

The MK3831 operates with the control register bits 0-6 hardwired as shown. This results in the operating frequency being fixed at 4.194304 MHz.

7	6	5	4	3	2	1	0
WP	0	1	0	0	0	1	0

CLOCK/CALENDAR/CONTROL BURST MODE

Address 31 Decimal of the Clock/Calendar/Control Address space specifies Burst Mode operation. In this mode, the 7 Clock/Calendar Registers and the Control Register may be consecutively read or written. Addresses above address 7 (Control Register) are non-existent; only addresses 0-7 are accessible.

RAM

The static RAM is contained in 24 writeable/readable registers, addressed consecutively in the RAM address space beginning at location 0.

RAM BURST MODE

Address 31 Decimal of the RAM address space specifies Burst Mode operation. In this mode, the 24 RAM registers may be consecutively read or written. Addresses above the maximum RAM address location are nonexistent and are not accessible.

REGISTER SUMMARY

A Register, Data Format summary is shown in Figure 4.

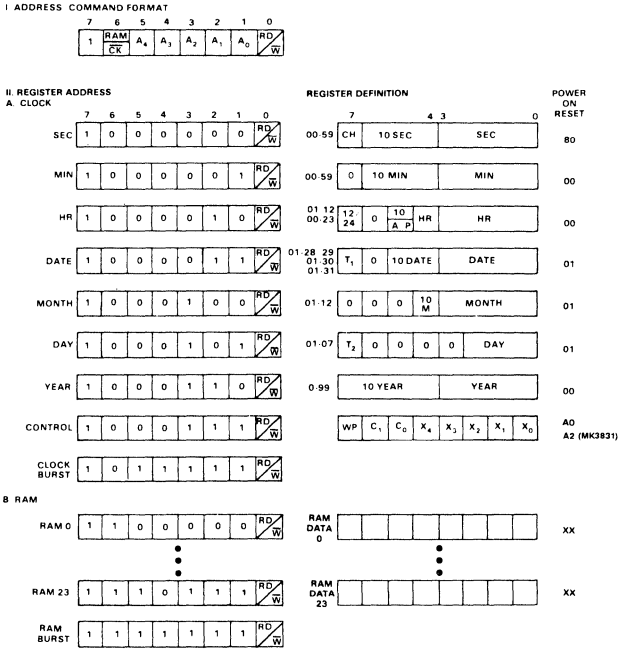
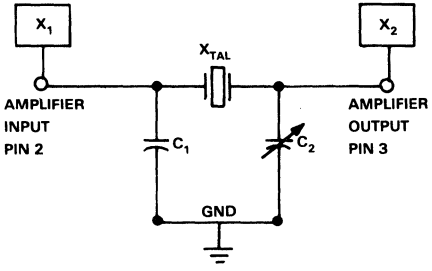


Figure 4. Microcomputer Clock/RAM Address/Command Register, Data Format Summary

CRYSTAL SELECTION

The wide frequency range of crystals that can be chosen for the Clock/RAM offers the user a large degree of flexibility. To aid in the selection of a suitable crystal, the following suggestions should be considered by the user. First, the MK3835 offers an output pin that will provide a system clock signal at either the crystal frequency, ½ the crystal frequency, or ¼ the crystal frequency. A system that requires a 4MHz clock initially may operate with an 8 MHz clock in the future. By applying an 8 MHz crystal to the Clock/RAM, a software change could provide the faster clock. Second, it is well known that, for a CMOS part, power dissipation will increase in direct proportion with frequency. Using a 1 MHz crystal and programming the CKO pin for 2048 Hz will cause the MK3835 to draw a minimum of power. (See Figures 9 and 10). The crystal connection is shown in Figure 5. If a generated clock signal is to be used as a time base, the connection is to Pin 2 (CKI) with Pin 3 left floating.



If it is desirable to "tune" the oscillator to a precise frequency, C_2 may be a variable capacitor.

C_2 should be in the range of $C_1 \leq C_2 \leq 2C$.

For C_L = 20 pf
C_1 is typically 30 pf.

Figure 5. Crystal Connection

Frequency Range	Specification
1 MHz — 8.4 MHz	Parallel resonance Fundamental mode C_L = 20 pf to 40 pf AT cut

Figure 6. Summary of Crystal Specifications

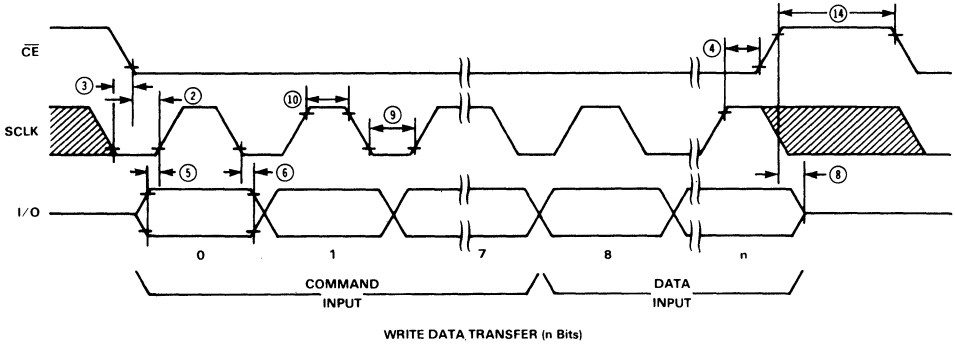


Figure 7. Input Timing Diagram

CRYSTAL FREQUENCY SELECTION

Table 2

Crystal Frequency Divider Mode	X4	X3	X2	X1	X0	f_{XTAL} (MHz) Crystal Frequency	Comments
Binary Mode	0	0	0	0	0	8.388608	Power on condition
	0	0	0	0	1	8.388608	
	0	0	0	1	0	4.194304	Hardwired in the MK3831
	0	0	0	1	1	4.194304	
	0	0	1	0	0	2.097152	
	0	0	1	0	1	2.097152	
	0	0	1	1	0	1.048576	
	0	0	1	1	1	0.032768	External Clock on CI only
Microprocessor Mode	0	1	0	0	0	8.000000	
	0	1	0	0	1	5.000000	
	0	1	0	1	0	4.000000	
	0	1	0	1	1	2.500000	
	0	1	1	0	0	2.000000	
	0	1	1	0	1	1.250000	
	0	1	1	1	0	1.000000	
	0	1	1	1	1	0.031250	External Clock on CI only
Baud Rate Mode	1	0	0	0	0	7.372800	
	1	0	0	0	1	7.372800	
	1	0	0	1	0	3.686400	
	1	0	0	1	1	3.686400	
	1	0	1	0	0	1.843200	
	1	0	1	0	1	1.843200	
	1	0	1	1	0	0.921600	
	1	0	1	1	1	0.028800	External Clock on CI only
Color Burst Mode	1	1	0	0	0	7.159040	
	1	1	0	0	1	7.159040	
	1	1	0	1	0	3.579520	
	1	1	0	1	1	3.579520	
	1	1	1	0	0	1.789760	
	1	1	1	0	1	1.789760	
	1	1	1	1	0	0.894880	
	1	1	1	1	1	0.027965	External Clock on CI only

CLOCK OUTPUT SELECTION

Table 3

C1	C0	CKO Output Frequency	Comments
0	0	f_{XTAL}	
0	1	$f_{XTAL} \div 2$	Power on condition. Hardwired in the MK3831 with CKO disabled.
1	0	$f_{XTAL} \div 4$	
1	1	=2048 Hz	Frequency applies for use with Binary Mode only. Other operating modes produce a CKO signal approximately equal to 2000 Hz.

ELECTRICAL SPECIFICATIONS MK3835N/MK3831N

ABSOLUTE MAXIMUM RATINGS*

Voltage on V_{CC} relative to GND ..... −0.5 V to + 7.5 V
Voltage on any pin ..... −0.5 V to + V_{CC} + .5
Temperature under bias ..... −50°C + 95°C
Storage Temperature ..... −55°C to + 125°C

*Stresses greater than those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.

DC OPERATING CONDITIONS

MK3835N-00/MK3831N-00 0°C ≤ T_A ≤ + 70°C, MK3835N-10/MK3831N-10 −40°C ≤ T_A ≤ +85°C

SYMBOL	PARAMETER	MIN	MAX	UNIT	NOTES
V_{CC}	Supply Voltage	4.5	5.5	V	1
V_{SB1}	Supply Voltage for Standby 1	3.5		V	1, 7
V_{SB2}	Supply Voltage for Standby 2	2.5		V	1, 7, 10

DC ELECTRICAL CHARACTERISTICS

V_{CC} = 5V ± 10%, MK3835N-00/MK3831N-00 0°C ≤ T_A ≤ + 70°, MK3835N-10/MK3831N-10 −40°C ≤ T_A ≤ + 85°C

SYMBOL	PARAMETER	MIN	MAX	UNIT	NOTES
I_{CC1}	Power Supply Current		6.0	mA	2
I_{CC2}	Power Supply Current		10.0	mA	3
I_{CC3}	Power Supply Current		2.0	mA	4
I_{CC4}	Power Supply Current		600	μA	5
I_{CC5}	Power Supply Current - MK3831		1.5	mA	9
I_{CC6}	Power Supply Current for Standby 1		200	μA	5
I_{CC7}	Power Supply Current for Standby 2		120	μA	5
I_{LI}	Input Leakage Current, SCLK and $\overline{CE}$	−1.0	1.0	μA	6
I_{LO}	Output Leakage Current, I/O Pin	−10.0	10.0	μA	6
V_{IH}	Logic "1" Voltage, All Inputs except X1,	2.2		V	1
V_{IL}	Logic "0" Voltage, All Inputs		0.8	V	1
V_{IHX1}	Logic "1" Voltage, X_1 Input	3.9			8
$V_{I/OH}$	Output Logic "1" Voltage, I/O Pin	2.4		V	1(I_{OH}=−100μA)
$V_{I/OL}$	Output Logic "0" Voltage, I/O pin		0.4	V	1(I_{OL}=3.8 mA)
V_{CKH}	Output Logic "1" Voltage, CKO pin	2.4		V	1(I_{OH}=1.0 mA)
V_{CKL}	Output Logic "0" Voltage, CKO pin		0.4	V	1(I_{OL}=5.0 mA)

NOTES:
1. All voltages referenced to GND.
2. Crystal/Clock Input frequency = 8.4 MHz, f_{CKO} = 4.2 MHz with 30 pf load.
3. Crystal/Clock Input frequency = 8.4 MHz, f_{CKO} = 8.4 MHz with 100 pf load.
4. Crystal/Clock input frequency = 8.4 MHz, f_{CKO} = 2084 Hz with 30 pf load.
5. Crystal/Clock Input frequency = 1MHz, f_{CKO} = 2048 Hz with a 30 pf load.
6. Measured with V_{CC} = 5.0V, 0 ≤ V_I ≤ 5.0V, outputs in high impedance state.
7. Applied to pin 8 to retain data during a power fault.
8. V_{IHX1} spec. applies only to the external clock input configuration.
9. MK3831 with Crystal/Clock Input frequency = 4.194 MHz.
10. 0°C ≤ T_A ≤ 50°C.

CAPACITANCE

T_A = 25°C

SYMBOL	PARAMETER	TYPICAL	MAX	UNIT	TEST CONDITION
C_I	Capacitance on Input Pin		10	pF	Note 1
$C_{I/O}$	Capacitance on I/O pin		12	pF	Note 1
C_X	Capacitance on X1/C1 and X2	2.5	5	pf	Note 1

NOTE:
Measured as $C = \frac{|\Delta t|}{\Delta V}$ with V = 3 V, and unmeasured pins grounded

AC ELECTRICAL CHARACTERISTICS

V_{CC} = 5V ± 10%, MK3835N-00/MK3831N-00 0°C ≤ T_A ≤ + 70°; MK3835N-10/MK3831N-10 −40°C ≤ T_A ≤ + 85°C

NUM	SYMBOL	PARAMETER	MIN	MAX	UNIT	NOTES
1	f_X	Crystal frequency	800	8400	kHz	
2	t_{CSS}	$\overline{CE}$ to SCLK↑ set up time	1.0		μs	1, 6
3	t_{SCS}	SCLK low set up time to $\overline{CE}$↓	40		ns	1, 6
4	t_{SCH}	SCLK↑ to $\overline{CE}$↑ hold time	1.0		μs	1, 5, 6
5	t_{DSS}	Input Data to SCLK↑ set up time	400		ns	1, 6
6	t_{SDH}	Input Data from SCLK↓ hold time	200		ns	1, 6
7	t_{SDD}	Output Data from SCLK↓ delay time		600	ns	1, 2, 3, 6
8	t_{CDZ}	$\overline{CE}$↑ to I/O high impedance		500	ns	1, 2, 3, 6
9	t_{SWL}	SCLK low time	1.95	∞	μs	
10	t_{SWH}	SCLK high time	1.95	∞	μs	
11	f_{SCLK}	SCLK frequency	DC	250	kHz	
12	t_{SR}, t_{SF}	SCLK Rise and Fall Time		1	μs	4, 6
13	t_{CR}, t_{CF}	CKO Rise and Fall Time		50	ns	4, 6
14	t_{CWH}	$\overline{CE}$ high time	2.0		μs	

NOTES:

1. Measured at V_{IH} = 2.0V or V_{IL} = 0.8V and 50 ns rise and fall times on inputs.
2. Measured at V_{OH} = 2.4V and V_{OL} = 0.4V.
3. Load Capacitance = 100 pF
4. t_r and t_f measured from 0.8V to 2.2V.
5. t_{SCH} must follow the last rising edge of S_{CLK} during a write cycle in order to allow time to complete a write to the internal register.
6. All voltages referenced to ground.

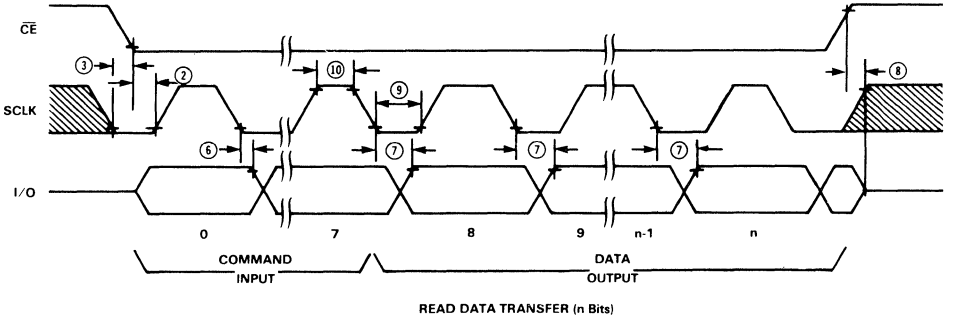


Figure 8. Output Timing Diagram

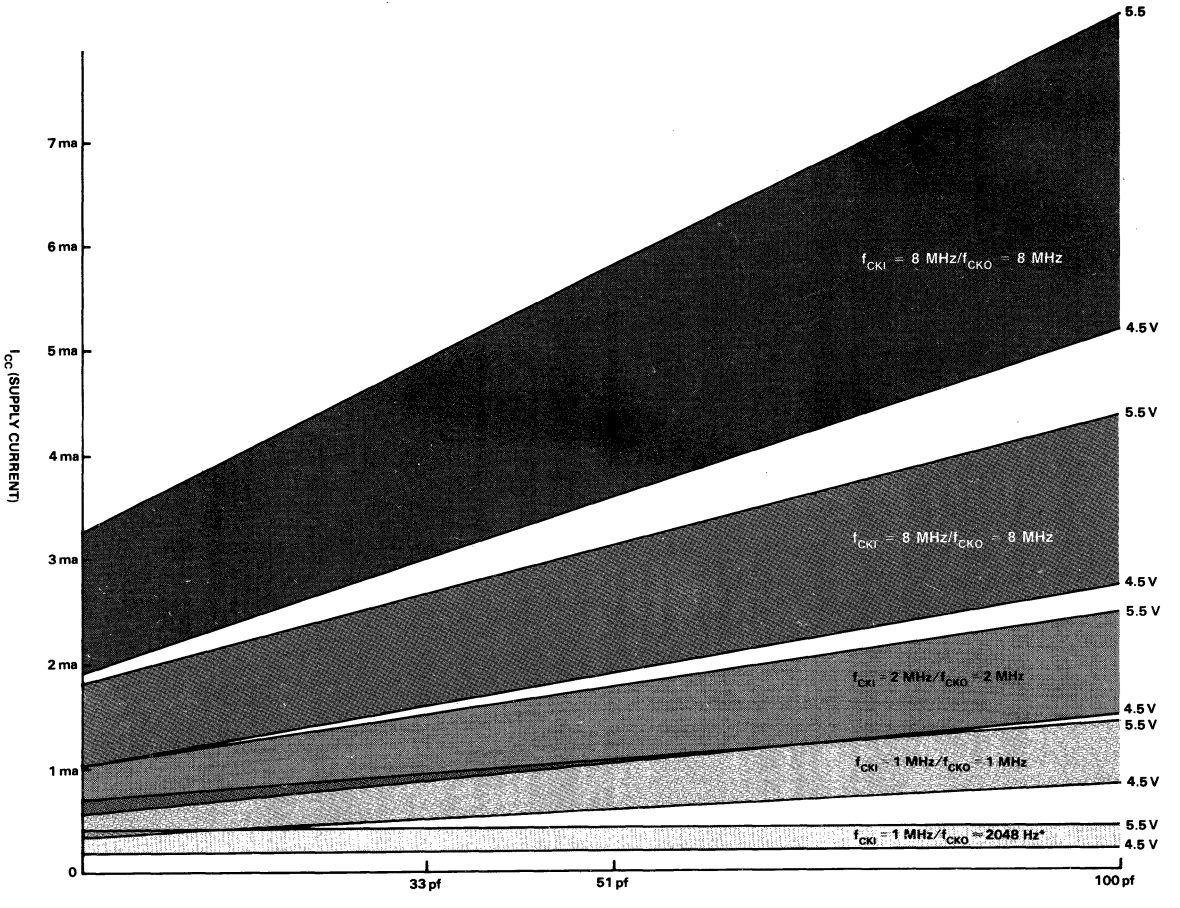


Figure 9. Typical Supply Current For Divide-By-One Mode Crystal Operation, T_A = 25°C

* This curve applies to the operating case where clock output selection is programmed for C1 = 1, CO = 1, and is included for comparison purposes to other operating modes.

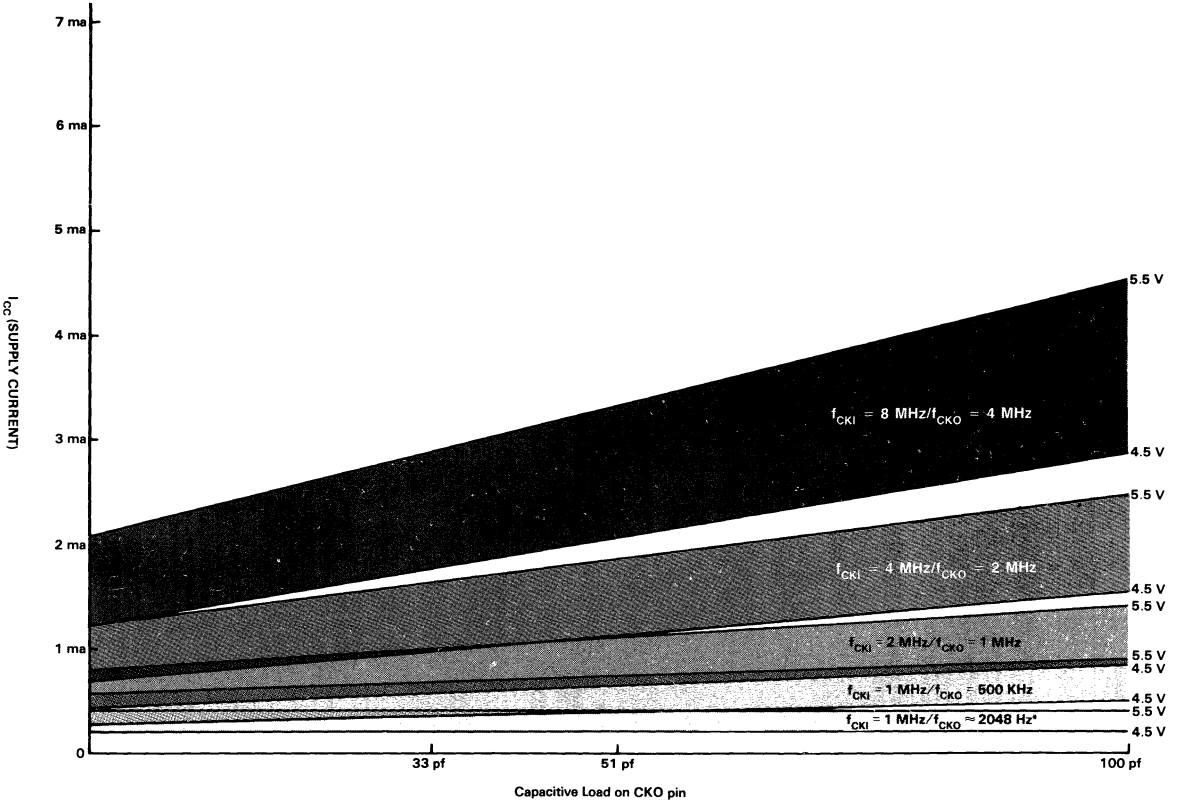
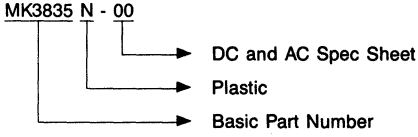


Figure 10. Typical Supply Current For Divide-By-Two Mode Crystal Operation, T_A = 25°C

* This curve applies to the operating case where clock output selection is programmed for C1 = 1, CO = 1, and is included for comparison purposes to other operating modes.

ORDERING INFORMATION

Device Order Number	V_{CC}	Temp	Package
MK3835N-00	5 V ± 10%	0°C - 70°C	Plastic
MK3835N-10	5 V ± 10%	−40°C - 85°C	Plastic
MK3831N-00	5 V ± 10%	0°C - 70°C	Plastic
MK3831N-10	5 V ± 10%	−40°C - 85°C	Plastic



CHAPTER 6 - CRT CONTROLLER

CRT CONTROLLERS SELECTION GUIDE

Part number	Description	Technology	Alt source	CLK freq. (MHz)	Page
EF9345	Single chip alphanumeric and semigraphic display processor - 25/21 rows of 40 or 80 char. - Multipage memory - Color and B/W	HMOS		40	6-3
EF9367	Graphic display coprocessor - Up to 512 × 1024 interlaced - 50/60 Hz color and B/W	HMOS		40	6-49
EF9369	"Palette" circuit for selection of 16 colors among 4096 - Compatible with all display circuits	HMOS		28	6-81
TS68483	High performance display coprocessor	HMOS		68	6-97
TS68494	Palette and serial port controller	HMOS 2		48	6-143

THOMSON COMPONENTS MOSTEK

EF9345

SINGLE CHIP SEMI-GRAPHIC DISPLAY PROCESSOR

The EF9345, new advanced color CRT controller, in conjunction with an additional standard memory package allows full implementation of the complete display control unit of a color or monochrome low-cost terminal, thus significantly reducing IC cost and PCB space.

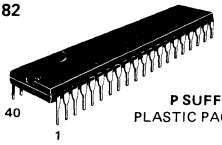
- Single chip low-cost color CRT controller
- TV standard compatible (50 Hz or 60 Hz)
- 2 screen formats :
  25 (or 21) rows of 40 characters
  25 (or 21) rows of 80 characters
- On-chip 128 alphanumeric and 128 semi-graphic character generator three standard options available for alphanumeric sets.
- Easy extension of user defined alphanumeric or semi-graphic sets (> 1 K characters).
- 40 characters/row attributes :
  Foreground and background color, double height, double width, blinking, reverse, underlining, conceal, insert, accentuation of lower case characters.
- 80 characters/row attributes :
  Underlining, blinking, reverse, color select.
- Programmable roll-up, roll down, and cursor display
- On-chip R, G, B, I video shift registers
- Easy synchronization with external video source : on-chip phase comparator.
- Address/data multiplexed bus directly compatible with standard microcomputers such as 6801, 6301, 8048, 8051.
- Addressing space : 16 K x 8 of general purpose private memory.
- Easy use of any low cost memory components : ROM, SRAM, DRAM
- Upward compatible with EF9340/41 chip set.

HMOS2

SINGLE CHIP SEMI-GRAPHIC DISPLAY PROCESSOR

CASES

CB-182

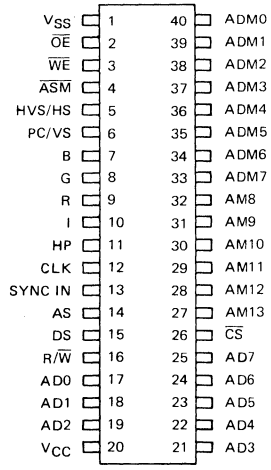


P SUFFIX
PLASTIC PACKAGE

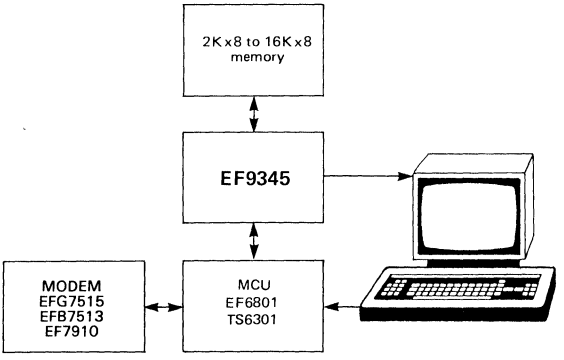
CB-521

FN SUFFIX
SURPICOP
(PLASTIC CHIP-CARRIER)

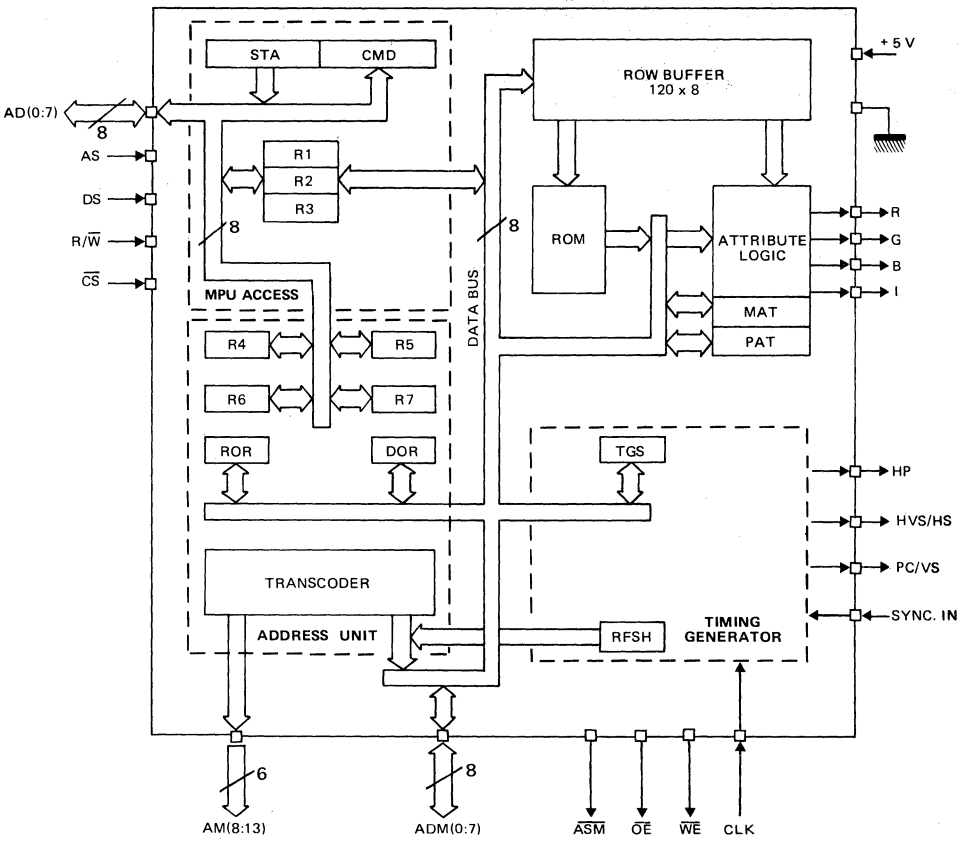
PIN ASSIGNMENT



TYPICAL APPLICATION
Low cost personal terminal



BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply voltage	V_{CC}*	- 0.3 to 7.0	V
Input voltage	V_{in}*	- 0.3 to 7.0	V
Operating temperature range	T_A	0 to 70	°C
Storage temperature range	T_{stg}	- 55 to 150	°C
Max power dissipation	P_{Dm}	0.75	W

*With respect to V_{SS}

Stresses above those hereby listed may cause permanent damage to the device. The ratings are stress ones only and functional operation of the device at these or any conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. Standard MOS circuits handling procedure should be used to avoid possible damage to the device.

ELECTRICAL OPERATING CHARACTERISTICS

(V_{CC} = 5.0 V ± 5 %, V_{SS} = 0, T_A = 0 to 70°C))

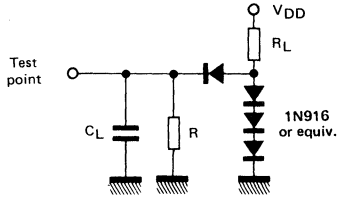
Characteristic	Symbol	Min	Typ	Max	Unit
Supply voltage	V_{CC}	4.75	5	5.25	V
Input low voltage	V_{IL}	- 0.3	–	0.8	V
Input high voltage CLK other inputs	V_{IH}	2.2 2	— —	V_{CC} V_{CC}	V —
Input leakage current	I_{in}	–	–	10	μA
Output high voltage (I_{load} = - 500 μA)	V_{OH}	2.4	–	–	V
Output low voltage I_{load} = 4 mA ; AD(0:7), ADM(0:7), AM(8:13) I_{load} = 1 mA ; other outputs	V_{OL}	0.4	–	–	V
Power dissipation	P_D	–	250	–	mW
Input capacitance	C_{in}	–	–	15	pF
Three state (off state) input current	I_{TSI}	–	–	10	μA

MEMORY INTERFACE

V_{CC} = 5.0 V ± 5 %, T_A = 0° to + 70°C.
Clock : f_{in} = 12 MHz ; duty cycle 40 to 60 % ; t_r, t_f < 5 ns
Reference levels : V_{IL} = 0.8 V and V_{IH} = 2 V, V_{OL} = 0.4 V and V_{OH} = 2.4 V.

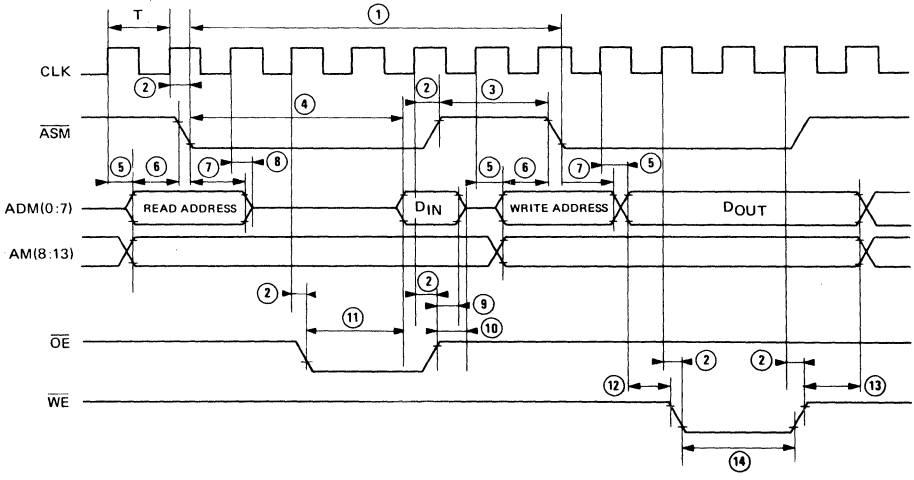
Ident. number	Characteristic	Symbol	Min	Typ	Max	Unit
1	Memory cycle time	t_{ELEL}	–	500	–	ns
2	Output delay time from CLK rising edge ($\overline{ASM}$, $\overline{OE}$, $\overline{WE}$)	t_D	–	–	60	ns
3	$\overline{ASM}$ high pulse width	t_{EHEL}	120	–	–	ns
4	Memory access time from $\overline{ASM}$ low	t_{ELDV}	–	–	290	ns
5	Output delay time from CLK rising edge (ADM(0:7), AM(8:13))	t_{DA}	–	–	80	ns
6	Address setup time to $\overline{ASM}$	t_{AVEL}	30	–	–	ns
7	Address hold time from $\overline{ASM}$	t_{ELAX}	55	–	–	ns
8	Address off time	t_{CLAZ}	–	–	80	ns
9	Memory hold time	t_{GHDX}	10	–	–	ns
10	Data off time from $\overline{OE}$	t_{OZ}	–	–	60	ns
11	Memory $\overline{OE}$ access time	t_{GLDV}	–	–	150	ns
12	Data setup time (write cycle)	t_{QVWL}	30	–	–	ns
13	Data hold time (write cycle)	t_{WHQX}	30	–	–	ns
14	$\overline{WE}$ pulse width	t_{WLWH}	110	–	–	ns

Test load



	ADM(0:7), AM(8:13) AD(0:7)	Other outputs
C	100 pF	50 pF
R_L	1 KΩ	3.3 KΩ
R	4.7 KΩ	4.7 KΩ

MEMORY INTERFACE TIMING DIAGRAM



MICROPROCESSOR INTERFACE

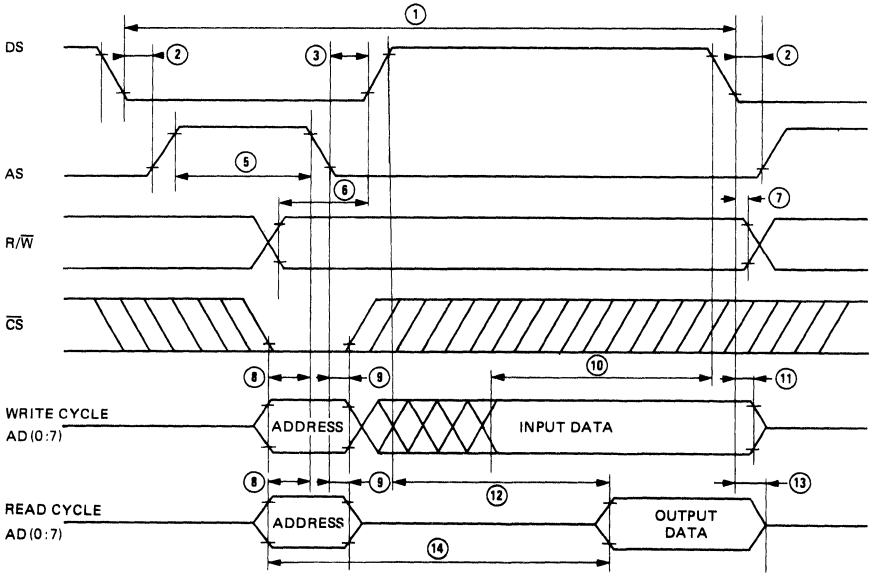
EF9345 is MOTEL compatible. It automatically selects the processor type by using AS input to latch to state of the DS input.
No external logic is needed to adapt bus control signals from most of the common multiplexed bus microprocessors.

EF9345	6801/6805CT	INTEL Family
	timing 1	timing 2
AS	AS	ALE
DS	DS, E, ϕ2	$\overline{RD}$
R/$\overline{W}$	R/$\overline{W}$	$\overline{WR}$

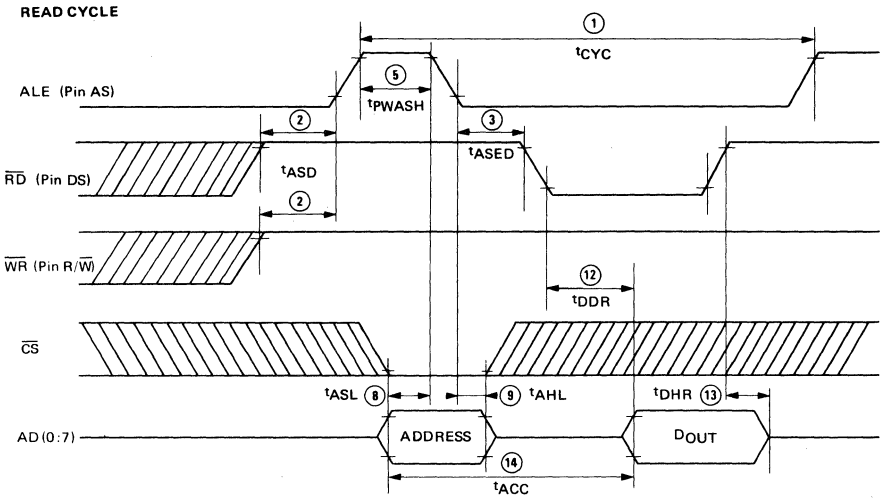
MICROPROCESSOR INTERFACE TIMING AD(0:7), AS, DS, R/$\overline{W}$, $\overline{CS}$
V_{CC} = 5.0 ± 5 %, T_A = 0° to + 70°C, C_L = 100 pF on AD(0:7)
Reference levels : V_{IL} = 0.8 V and V_{IH} = 2 V on all inputs ; V_{OL} = 0.4 V and V_{OH} = 2.4 V on all outputs.

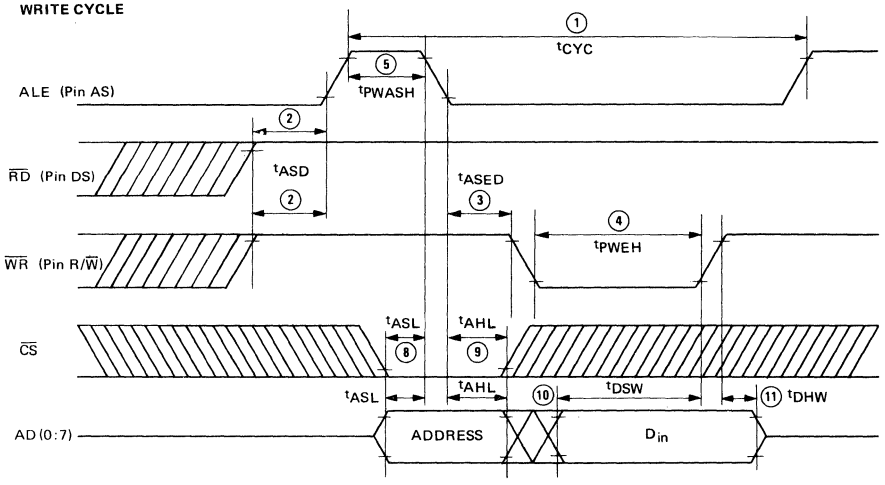
Ident. number	Characteristic	Symbol	Min	Typ	Max	Unit
1	Cycle time	t_{CYC}	400	–	–	ns
2	DS low to AS high (timing 1) DS high or R/$\overline{W}$ high to AS high (timing 2)	t_{ASD}	30	–	–	ns
3	AS low to DS high (timing 1) AS low to DS low or R/$\overline{W}$ low (timing 2)	t_{ASED}	30	–	–	ns
4	Write pulse width	t_{PWEH}	200	–	–	ns
5	AS pulse width	t_{PWASH}	100	–	–	ns
6	R/$\overline{W}$ to DS setup time (timing 1)	t_{RWS}	100	–	–	ns
7	R/$\overline{W}$ to DS hold time (timing 1)	t_{RWH}	10	–	–	ns
8	Address and $\overline{CS}$ setup time	t_{ASL}	20	–	–	ns
9	Address and $\overline{CS}$ hold time	t_{AHL}	20	–	–	ns
10	Data setup time (write cycle)	t_{DSW}	100	–	–	ns
11	Data hold time (write cycle)	t_{DHW}	10	–	–	ns
12	Data access time from DS (read cycle)	t_{DDR}	–	–	150	ns
13	DS inactive to high impedance state time (read cycle)	t_{DHR}	10	–	80	ns
14	Address to data valid access time	t_{ACC}	–	–	300	ns

MICROPROCESSOR INTERFACE TIMING DIAGRAM 1
(6801 / 6805CT)



MICROPROCESSOR INTERFACE TIMING DIAGRAM 2 (INTEL type)



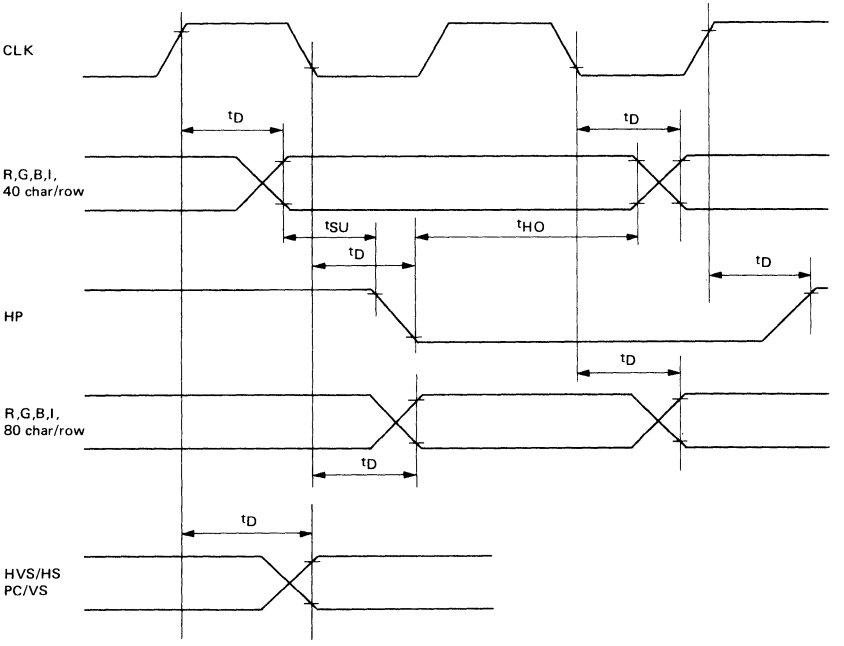


VIDEO INTERFACE R, G, B, I, HP, HVS / HS, PC / VS

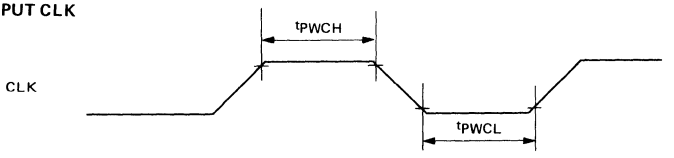
V_{CC} = 5.0 V ± 5 %, T_A = 0° to + 70°C, CLK duty cycle = 50 %, C_L = 50 pF

Reference levels : V_{IL} = 0.8 V and V_{IH} = 2.2 V on CLK input. V_{OL} = 0.4 V and V_{OH} = 2.4 V on all outputs.

Characteristic	Symbol	Min	Typ	Max	Unit
Set up time R, G, B, I to HP	t_{SU}	10	–	–	ns
Hold time R, G, B, I from HP	t_{HO}	50	–	–	ns
Output delay from CLK edge	t_D	–	–	60	ns



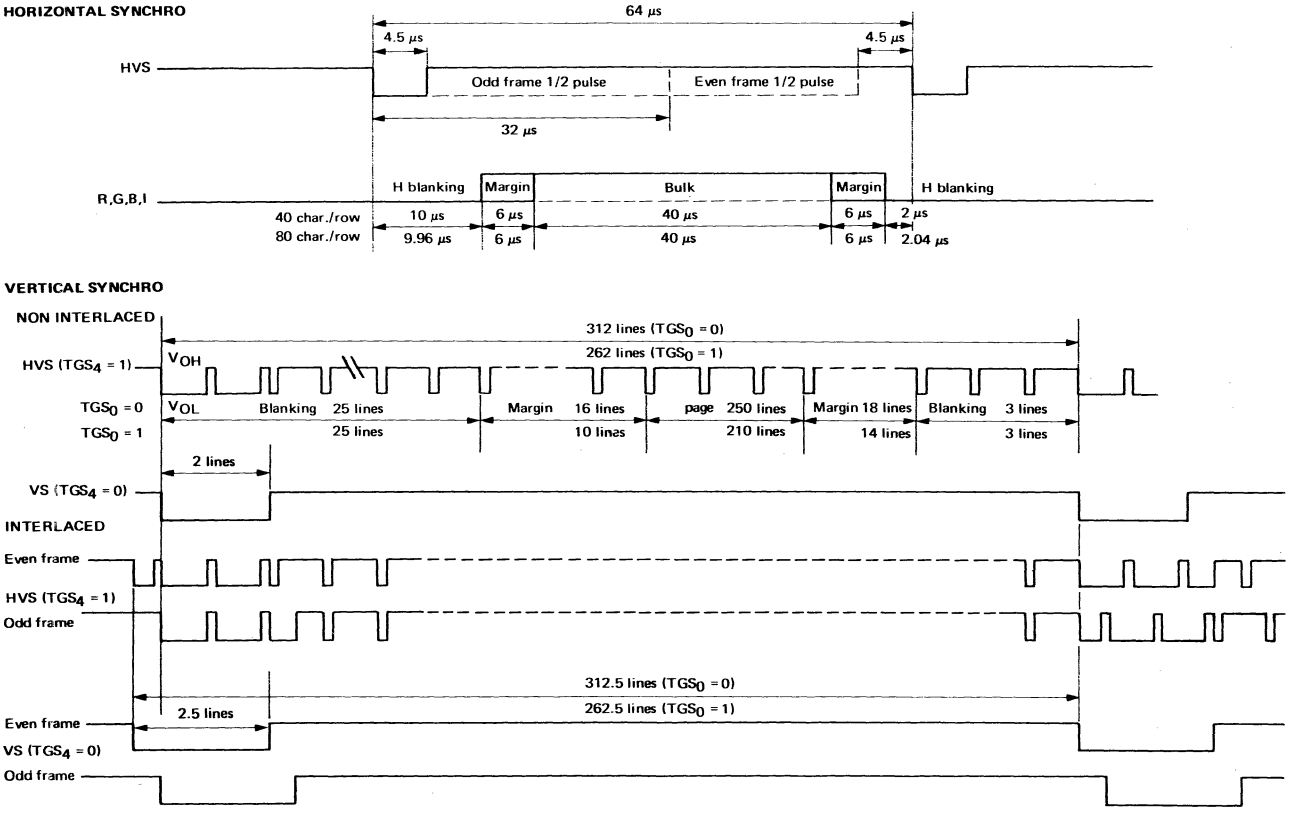
INPUT CLK



Reference level V_{IL} = 0.8 V, V_{IH} = 2.0 V

Characteristic	Symbol	Min	Unit
CLK high pulse width	t_{PWCH}	30	ns
CLK low pulse width	t_{PWCL}	30	ns

VERTICAL AND HORIZONTAL SYNCHRONIZATION OUTPUTS (CLK = 12 MHz)



EF9345 PIN DESCRIPTION

All the input/output pins are TTL compatible.

MICROPROCESSOR INTERFACE

NAME	PIN TYPE	N°	FUNCTION	DESCRIPTION
AD(0:7)	I/O	17-19 21-25	Multiplexed Address/Data bus	These 8 bidirectional pins provide communication with the microprocessor system bus.
AS	I	14	Address Strobe	The falling edge of this control signal latches the address on the AD(0:7) lines, the state of the Data Strobe (DS) and Chip Select ($\overline{CS}$) into the chip.
DS	I	15	Data Strobe	When this input is strobed high by AS, the output buffers are selected while DS is low for a read cycle ($R/\overline{W} = 1$). In write cycle, data present on AD(0:7) lines are strobed by $R/\overline{W}$ low (See Timing Diagram 2). When this input is strobed low by AS, $R/\overline{W}$ gives the direction of data transfer on AD(0:7) bus. DS high strobes the data to be written during a write cycle ($R/\overline{W} = 0$) or enables the output buffers during a read cycle ($R/\overline{W} = 1$). (See Timing Diagram 1).
$R/\overline{W}$	I	16	Read / Write	This input determines whether the internal registers get written or read. A write is active low ("0").
$\overline{CS}$	I	26	Chip Select	The EF9345 is selected when this input is strobed low by AS.

MEMORY INTERFACE

ADM(0:7)	I/O	40-33	Multiplexed Address/Data bus	Lower 8 bits of memory address appear on the bus when $\overline{ASM}$ is high. It then becomes the data bus when $\overline{ASM}$ is low.
AM(8:13)	O	32-27	Memory Address bus	These 6 pins provide the high order bits of the memory address.
$\overline{OE}$	O	2	Output Enable	When low, this output selects the memory data output buffers.
$\overline{WE}$	O	3	Write Enable	This output determines whether the memory gets read or written. A write is active low ("0").
$\overline{ASM}$	O	4	Memory Address Strobe	This signal cycles continuously. Address can be latched on its falling edge.

VIDEO INTERFACE

NAME	PIN TYPE	N°	FUNCTION	DESCRIPTION
R G B	O O O	7 8 9	Red Green Blue	These outputs deliver the video signal. They are low during the vertical and horizontal blanking intervals.
I	O	10	Insert	This active high output allows to insert R:G:B in an external video signal for captioning purposes, for example. It can also be used as a general purpose attribute or color.
HVS/HS	O	5	Sync. Out	This output delivers either the composite synchro (bit $TGS_4 = 1$) or the horizontal synchro signal (bit $TGS_4 = 0$).
PC/VS	O	6	Phase Comparator/ Vertical Sync	When $TGS_4 = 1$, this signal is the phase comparator output. When $TGS_4 = 0$, this output delivers the vertical synchro signal.
SYNC IN	I	13	Synchro In	This input allows vertical and/or horizontal synchronizing the EF9345 on an external signal. It must be grounded if not used.
HP	O	11	Video Clock	This output delivers a 4 MHz clock phased with the R, G, B, I signals.

OTHER PINS

CLK	I	12	Clock Input	External TTL clock input. (Nominal value : 12 MHz, duty cycle : 50 %).
V_{SS}	S	1	Power Supply	Ground.
V_{CC}	S	20	Power Supply	+ 5 V.

The EF9345 is a low cost, semigraphic, CRT controller.

It is optimized for use with a low cost, monochrome or color TV type CRT (64 μs per line, 50 or 60 Hz refresh frequency).

The EF9345 displays up to 25 rows of 40 characters or 25 rows of 80 characters.

The on-chip character generator provides a 128 standard, 5 x 7, character set and standard semigraphic sets.

More user definable (8 x 10) alphanumeric or semigraphic sets may be mapped in the 16K x 8 private memory addressing space.

These user definable sets are available only in 40 characters per row format.

MICROPROCESSOR INTERFACE

The EF9345 provides an 8-bit, address/data multiplexed, microprocessor interface.

It is directly compatible with popular (6801, 6805CT, 8048, 8051, 8035...) microprocessors.

REGISTERS

The microprocessor directly accesses 8 registers :

- R0 : Command/status register
- R1, R2, R3 : Data registers
- R4, R5 } : Each of these register pairs points into the R6, R7 } private memory.

Through these registers, the microprocessor indirectly accesses the private memory and 5 more registers :

- ROR, DOR : Base address of displayed page memory and of used external character generators.
- PAT, MAT, TGS : Used to select the page attributes and format, and to program the timing generator option.

PRIVATE MEMORY

The user may partition the 16K x 8 private memory addressing space between :

- pages of character codes (2K x 8 or 3K x 8),
- external character generators,
- general purpose user area.

Many types of memory components are suitable :

- ROM, DRAM or SRAM,
- 2K x 8, 8K x 8, 16K x 4 organizations,
- Modest 500 ns cycle time and 250 ns access time is required.

40 CHARACTERS PER ROW : CHARACTER CODE FORMATS AND ATTRIBUTES

Once the 40 characters per row format has been selected, one character code format out of three must be chosen :

- 24-bit fixed format :
  All the attributes are provided in parallel.
- 8/24-bit compressed format :
  All the attributes are latched.
- 16-bit fixed format :
  Some parallel attributes, others are latched.

The 16-bit fixed format is compatible with EF9340/41 CRT controller.

Character attributes provided :

- Background and foreground color (3 bits each),
- Double height, double width,
- Blinking,
- Reverse,
- Underlining,
- Conceal,
- Insert,
- Accentuation of lower case characters
- 3 x 100 user definable character generator in memory
- 8 x 100 semi-graphic quadrichrome characters.

80 CHARACTERS PER ROW FORMAT : CHARACTER CODE FORMAT AND ATTRIBUTES

Two character code formats are provided :

- Long (12 bits) with 4 parallel attributes :
  - Blinking,
  - Underlining,
  - Reverse,
  - Color select.
- Short (8 bits) : no attributes.

TIMING GENERATOR

The whole timing is derived from a 12 MHz main clock input.

The RGB outputs are shifted at 8 MHz for the 40 character/row format and at 12 MHz for the 80 character/row.

Besides, the user may select :

- 50 Hz or 60 Hz vertical sync. frequency,
- Interlaced or not,
- Separated or composite vertical and horizontal sync. outputs.

Furthermore, a composite sync. input allows, when it is required :

- An on-chip vertical resynchronization,
- An on-chip crude horizontal resynchronization,
- An off-chip high performance horizontal resynchronization by use of a simple external VCXO controlled by the on-chip phase comparator.

LOGICAL AND PHYSICAL ADDRESSING

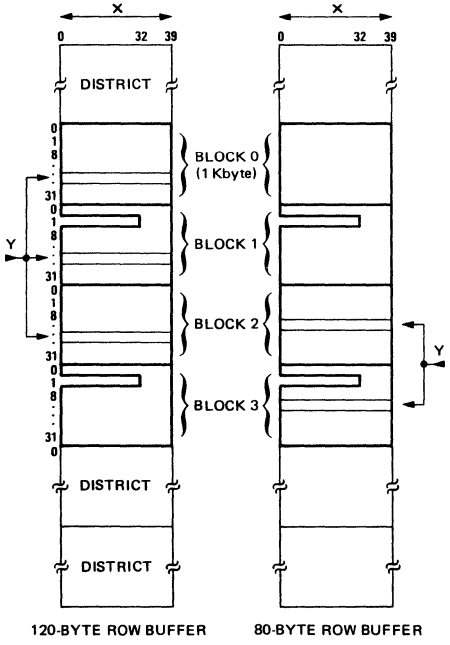
The physical 16-Kbyte addressing space is logically partitioned by EF9345 into 40-byte **buffers** (Figure 1). More precisely, a logical address is given by an X, Y, Z triplet where :

- X = (0 to 39) points to a byte inside a buffer,
- Y = (0, 1 ; 8 to 31) points to a buffer inside a 1 Kbyte block,
- Z = (0 to 15) points to a **block.**

Obviously, 1 K = 2^{10} = 1024 cannot be exactly divided by 40. Consequently, any block holds 25 full buffers and a 24-byte remainder. Provided that the physical memory is a multiple of 2 Kbytes, the remainders are paired in such a way as to make available :

- a full buffer (Y = 1) in each even block,
- a partial buffer (Y = 1 ; X = 32 to 39) in each odd block.

FIGURE 1 – MEMORY ROW BUFFER



- Row buffers lay inside a district.
- At two or three successive block addresses (modulo 4).
- First block address is even.

POINTERS

Each X, Y and Z component of a logical address is binary encoded and packed in two 8-bit registers. Such a register pair is a pointer (Figure 2). EF9345 contains two pointers :

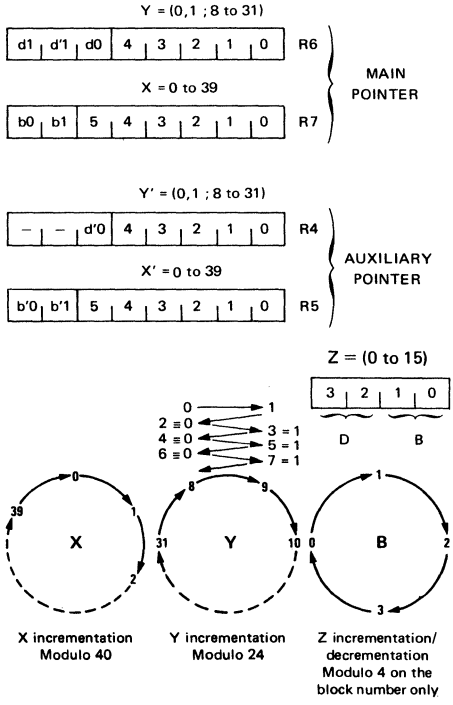
- R4, R5 : **auxiliary pointer,**
- R6, R7 : **main pointer.**

R5 and R7 have the same format. Each one holds an X component and the two LSB's of a Z component. This packing induces a partitioning of Z in 4 **districts** of 4 blocks each.

R5, R7 points to a block number in a district. R4 and R6 have a slightly different format : Each one holds a Y component and the LSB of the district number. But R6 holds both district MSB.

Figure 3 gives the logical to physical address transcoding scheme performed on chip.

FIGURE 2 – POINTER AUTO INCREMENTATION



DATA STRUCTURES IN MEMORY

A **page** is a data structure displayable on the screen up to 25 rows of characters. According to the character code format, each row on the screen is associated with 2 (or 3) 40-byte buffers. This set of 2 (or 3) buffers constitutes a row buffer (Figure 1). The buffers belonging to a **row buffer** must meet the following requirements :

- they have the same Y address,
- they have the same district number,
- they lie at 2 (or 3) successive (modulo 4) block addresses in their common district.

Consequently, a row buffer is defined by its first buffer address and its format.
A page is a set of successive row buffers :

- with the same format,
- with the same district number,
- with the same block address of first buffer. This block address must be even.
- lying at successive (modulo 24) Y addresses.

Consequently, a page should not cross a district boundary. General purpose memory area may be used but should respect the buffer or row buffer structure. See Figure 2 for pointer incrementation implied by these data structures.

MEMORY TIME SHARING (See Figure 4)

The memory interface provides a 500 ns cycle time. That is to say a 2 Mbyte/s memory bandwidth. This bandwidth is shared between :

- reading a row buffer from memory to load the internal row buffer (up to 120 bytes once each row),
- reading user defined characters slices from memory (1 byte each μs),
- indirect microprocessor read or write operation,
- refresh cycles to allow DRAM use, with no overhead.

A fixed allocation scheme implements the sharing.

Notes on Figure 4.

1. Dummy cycles are read cycles at dummy addresses.
2. RFSH cycles are read cycles performed by an 8-bit auto-incrementing counter. Low order address byte ADM(0:7) cycles through its 256 states in less than 1 ms.
3. The microprocessor may indirectly access the memory once every μs, except during the first and the last line of a row, when the internal buffer must be reloaded.

During these lines, no microprocessor access is provided for 104 μs ; this hold too when no user defined character slices are addressed.

FIGURE 4 - MEMORY CYCLE ALLOCATION

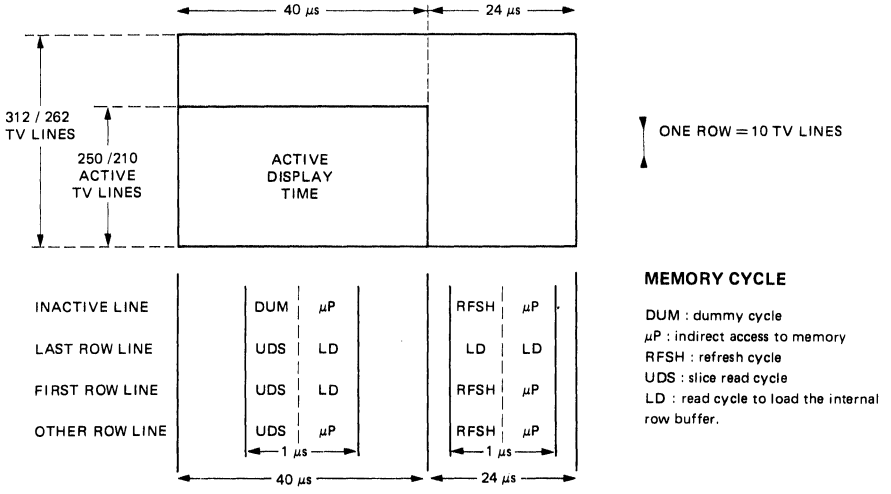
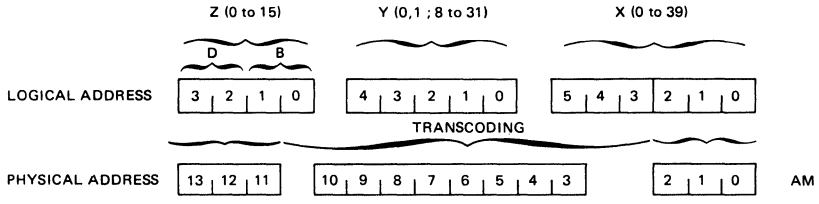


FIGURE 3 – LOGICAL TO PHYSICAL ADDRESS TRANSCODING PERFORMED ON-CHIP



X and Y CONDITION			PHYSICAL ADDRESS AM (3:10)							
			10	9	8	7	6	5	4	3
Y ⩾ 8	X5 = 0		b0	Y4	Y3	Y2	Y1	Y0	X4	X3
	X5 = 1		b0	0	0	Y2	Y1	Y0	Y4	Y3
Y < 8	Y0 = 0		b0	0	0	X5	X4	X3	0	0
	Y0 = 1	b0 = 0	X3	0	0	1	$\overline{X5}$	$\overline{X4}$	0	0
		b0 = 1	1	0	0	1	$\overline{X5}$	$\overline{X4}$	0	0

SCREEN FORMAT AND ATTRIBUTES

The screen format and attributes are programmed through 5 indirectly accessible registers : **ROR, TGS, PAT, MAT** and **DOR.** IND command allows accessing these registers. TGS is also used to select the timing generator options (see Screen Format Table).

ROW AND CHARACTER CODE FORMAT PAT_7 ; $TGS_{(6:7)}$

Two row formats and 5 character code formats are available but cannot be mixed in a given screen. DOR register interpretation is completely row format dependant and is discussed in the corresponding 40 char/row and 80 char/row section.

SCREEN PARTITION - PAGE POINTER ROR (See top of the Screen Format Table)

The screen is partitionned into 3 areas :

- the margin,
- the service row,
- the bulk of remaining rows.

$MAT_{(0:3)}$ declares the color of the margin and the value i_M of its insert attribute.
ROR register points to the page to be displayed and gives the 3 MSB's of the Z address : $Z_0 = 0$ implicitly ; the page block address must be even. YOR gives the first row buffer to be displayed at the top of the bulk area. The next row buffers to be displayed are fetched sequentially by incrementing the Y address (modulo 24). This address never gets out of the origin block. Incrementation of YOR by the microprocessor yields a roll up.

SERVICE ROW : TGS_5 - PAT_0

The service row is displayed for 10 TV lines on top of the screen and does not roll. Following TGS_5, it is fetched from the origin block at either Y = 0 or Y = 1. The Y = 1 is a partial row buffer. It can be used only with variable 40 char./row and an 8 byte attribute file. The service row may be disabled by $PAT_0 = 0$; it is then displayed as a margin extension.

BULK : TGS_0 ; $PAT_{(1;2)}$; MAT_7

It is displayed after the service row for 200 or 240 TV lines according to TGS_0. Each row buffer is usually displayed for 10 TV lines. However, $MAT_7 = 1$ doubles this figure. Then every character appears in double height (double height characters are quadrupled).

$PAT_1 = 0$ and/or $PAT_2 = 0$ disables respectively the upper 120 lines and/or the lower 80/120 lines of the bulk.
When disabled, the corresponding TV lines are displayed as a margin extension.

CURSOR $MAT_{(4:6)}$

To be displayed with the cursor attributes, a character must be pointed by the main pointer (R6, R7) and MAT_6 must be set. The cursor attributes are given by $MAT_{(4:5)}$:

- **Complementation :**
  the R, G and B of each pixel is logically negated.
  R, G, B ⟶ $\overline{R}$, $\overline{G}$, $\overline{B}$

- **Underline :**
  the underline attribute of this character is negated.
- **Flash :**
  the character is periodically displayed with, then without, its cursor attributes (50 % / 50 % ; ≃ 1 Hz).

FLASH ENABLE (PAT_6) - CONCEAL ENABLE (PAT_3)

Any character flashing attribute is a "don't care" when $PAT_6 = 0$. When $PAT_6 = 1$, a character flashes if its flashing attribute is set. It is then periodically displayed as a space (50 % / 50 % ; ≃ 0.5 Hz).

PAT_3 is a "don't care" for 80 char./row formats. When any 40 char./row format is in use :

- if $PAT_3 = 0$, the conceal attribute of any character is a don't care,
- if $PAT_3 = 1$, the conceal attribute of each character is interpreted : a concealed character appears as a space on the screen.

INSERT MODES : $PAT_{(4:5)}$

During **retrace, margin and extended margin periods,** the I output pin delivers the value of the insert margin attribute. $I = i_M = MAT_4$.

During **active line** period, the I output state is controlled by the Insert Mode and i, the insert attribute of each character. The I output pin may have several uses : (See figure below) :

- As a margin/active area signal in the active area mark mode.
- As a character per character marker signal in the **character mark** mode.
- As a video mixing signal in the two remaining modes, provided that EF9345 has been vertically and horizontally synchronized with an external video source : the I pin allows mixing RGB outputs (I=1) and the external video signal (I=0). This mixing can be achieve by switching or ORing. It may occur for the complete character window (**Boxing Mode**) or only for the foreground pixels (**Inlay Mode**).

Video outputs during active periods

INSERT MODE	CHAR. LEVEL		OUTPUTS	
	i	PIXELS (1)	I	R,G,B (2)
ACTIVE AREA MARK	–	–	1	X
CHARACTER MARK	0	–	0	X
	1	–	1	X
BOXING	0	–	0	BLACK
	1	–	1	X
INLAY	0	–	0	BLACK
	1	BACKGND	0	BLACK
		FOREGND	1	X

NOTES :
(1) PIXEL TYPE
– : Don't care

FOREGND : A foreground pixel is :

- Any pixel of a quadrichrome character,
- A pixel of a bichrome character generated from a "1" in the character generator cell.

(2) RGB OUTPUTS

X : Not affected

BLACK : forced to low level.

TIMING GENERATOR OPTIONS : $TGS_{(0:4)}$

$TGS_{(0:1)}$ select the number of lines per frame :

TGS_1	TGS_0	LINES	
0	0	312	NON INTERLACED
0	1	262	
1	0	312.5	INTERLACED
1	1	262.5	

The composite incoming SYNC IN signal is separated into 2 internal signals :

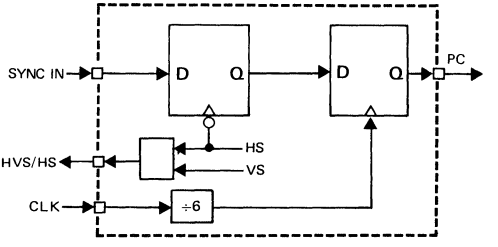
- Vertical Synchronization In (VSI),
- Horizontal Synchronization In (HSI).

TGS_3 enables VSI to reset the internal line count. SYNC IN input is sampled at the beginning of the active area of each line. When the sample transits from 1 to 0, the line count is reset at the end of the current line.

TGS_2 enables HSI to control an internal digital phase lock loop. HSI and on-chip generated HS Out are considered as in phase if their leading edges match at ± 1 clock period. When they are out of phase, the line period is lengthened by 1 clock period (≃ 80 ns).

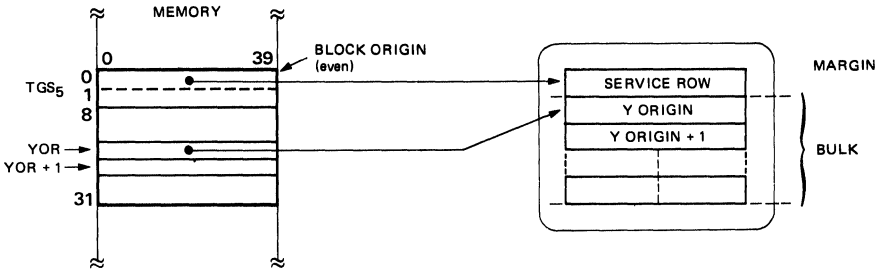
TGS_4 controls the SYNC OUT pins configuration :

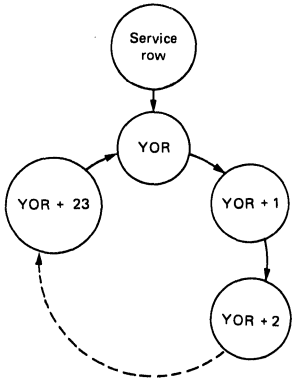
TGS_4	HVS / HS	PC / VS
1	COMPOSITE SYNC	PC
0	H SYNC OUT	V SYNC OUT

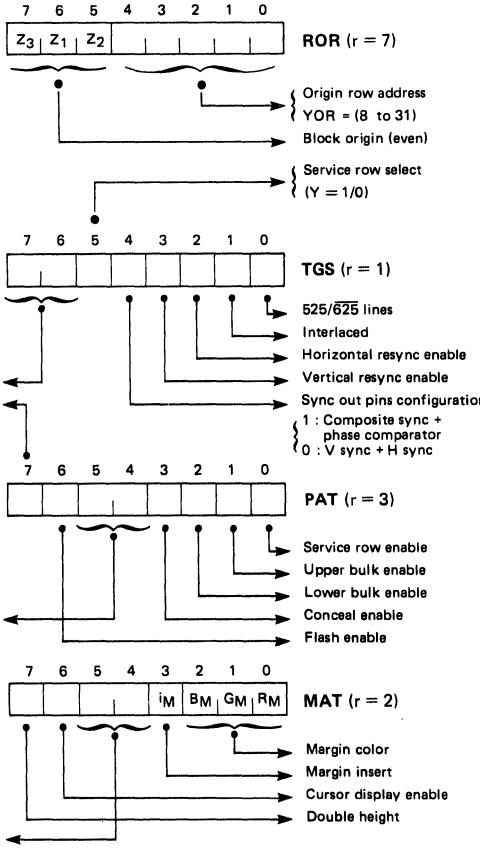


PC is the output of the on-chip phase comparator.
An external VCXO allows a smoother horizontal phase lock than the internal scheme.

SCREEN FORMAT TABLE







CHAR CODE	PAT_7	TGS_7	TGS_6
40 CHAR LONG	0	0	0
40 CHAR VAR	0	0	1
40 CHAR SHORT	1	0	0
80 CHAR LONG	0	1	1
80 CHAR SHORT	0	1	0

INSERT MODE	PAT_5	PAT_4
INLAY	0	0
BOXING	0	1
CHARACTER MARK	1	0
ACTIVE AREA MARK	1	1

CURSOR DISPLAY MODE	MAT_5	MAT_4
FIXED COMPLEMENTED	0	0
FLASH COMPLEMENTED	1	0
FIXED UNDERLINED	0	1
FLASH UNDERLINED	1	1

NOTE : PROGRAMMING BIT VALUE
1 = True
0 = False

40 CHAR/ROW CHARACTER CODES

To display pages in 40 character per row format, one out of three character code formats must be selected :

- Fixed long (24 bits) code : all parallel attributes.
- Fixed short (16 bits) code : mix of parallel and latched attributes.
- Variable (8/24 bits) code : all latched attributes.

Fixed short and variable codes are translated into fixed long codes by EF9345 during the internal row buffer loading process. The choice of the character code format is obviously a display flexibility/memory size trade off, left up to the user.

FIXED LONG CODES

This is the basic 40 char./row code. Each 8 pixels x 10 lines character window, on the screen is associated with a 3-byte code in memory, namely the C, B and A bytes (Figure 5). A row on the screen is associated with a 120 byte row buffer in memory.

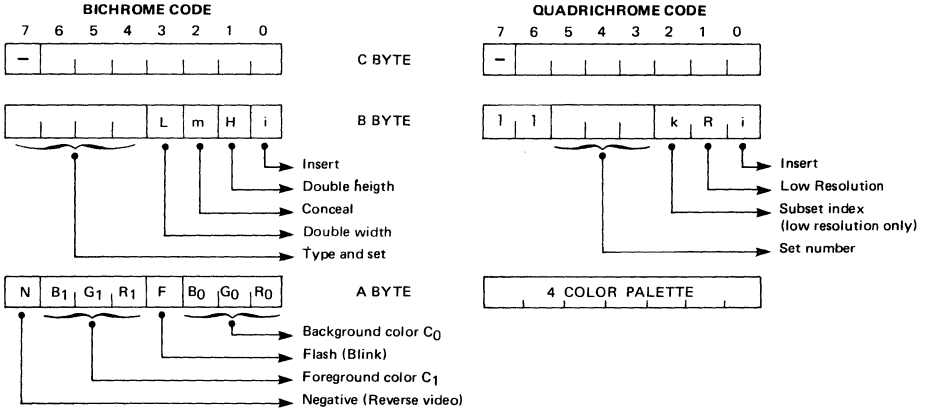
3-byte code structure

1. C7 is a don't care. Up to 128 characters may be addressed in each set. Each user definable set holds only 100 characters : C byte value ranges from 00 to 03 and 20 to 7F (hexa).

2. B(4:7) give the type and set number of the character.

3. All the bichrome characters have the same attributes except that alphanumerics may be underlined, semi-graphics cannot. Accentuated alphanumerics allow orthogonal accentuating of any one of the 32 lower case ROM characters with any of 8 accents (see Figure 19).

FIGURE 5 – 40 CHAR/ROW FIXED LONG CODES



TYPE AND SET CODE : B(4:7)				NUMBER OF CHARACTER PER SET	SET NAME	SET TYPE		CELL LOCATION
7	6	5	4	C(0:6)				
0	0	1	0	128 standard mosaics	G_{10}	SEMI-GR.	BICHROME	ON-CHIP ROM
		1	1	32 strokes	G_{11}			
	0	0	UNDERLINE	128 alphanumerics	G_0	ALPHA		
	1	0		Accentuated lower case alpha	G_{20}			
		1			G_{21}			
1	0	0		100 alpha UDS	G'_0			EXTERNAL MEMORY
	0	1	0	100 semi-graphic UDS	G'_{10}	SEMI-GR.		
		1	1	100 semi-graphic UDS	G'_{11}			
	1	X	X	8 sets of 100 quadrichrome character	Q_0 to Q_7	QUADRICHROME		

NOTA : Programming bit value
1 = True
0 = False

4. Bichrome and quadrichrome characters use two different coloring schemes.

For bichrome characters, character code byte A defines a two color set by giving directly two color values (Figure 6). The negative attribute exchanges the two values. Each bit of the slice byte selects one color value out of two.

The "A" byte in a quadrichrome character code defines an ordered 4 color set (Figure 7). When more than 4 bits are set, higher ranking bits are ignored. When less than 4 bits are set, the color set is completed with implicit "white" value. The slice byte is shifted 2 bits at once at half the dot frequency (≃ 4 MHz). Each bit pair designates one color out of the 4 color sets.

Quadrichrome characters allow displaying up to 4 different colors (instead of 2) in any 8 x 10 window at the penalty of an halved horizontal resolution.
By programming the R attribute in byte B, one may chose to keep the full vertical resolution (1 slice per line) or to halve it (each slice is repeated twice). In any case, it is possible to change the color set freely from window to window and to mix freely all the character types. So, fairly complex pictures may be displayed at low memory cost.

FIGURE 6 – COLORING WITH BICHROME CHARACTERS

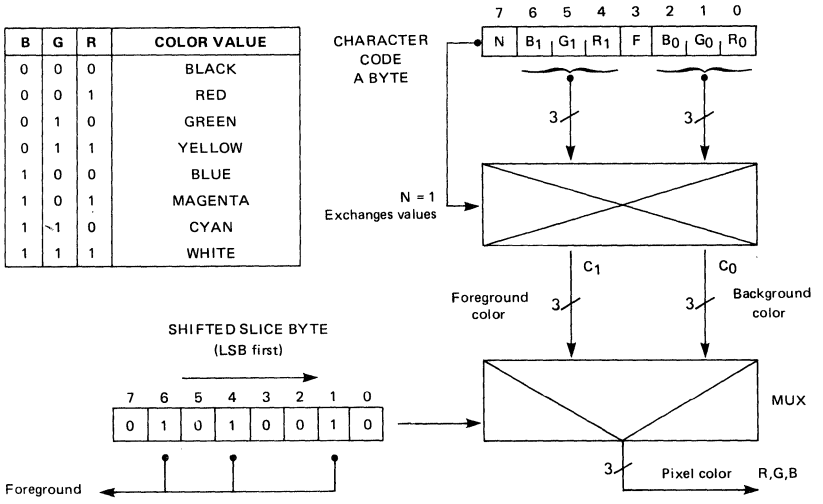
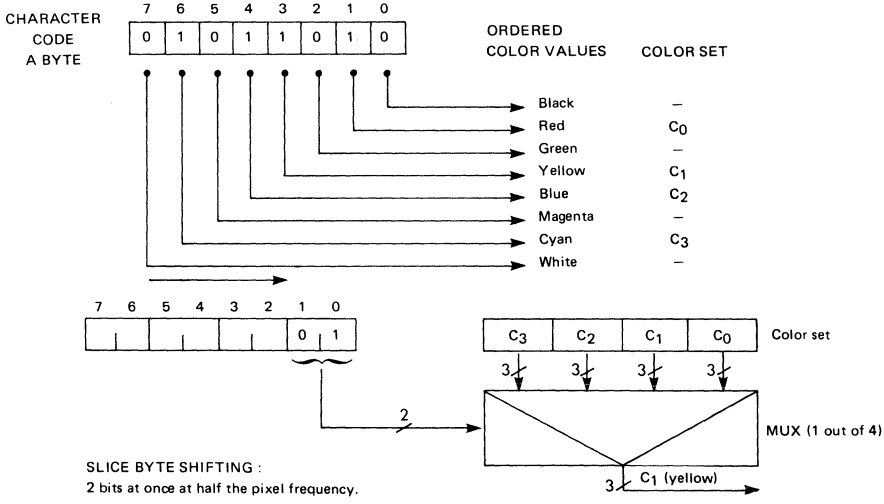


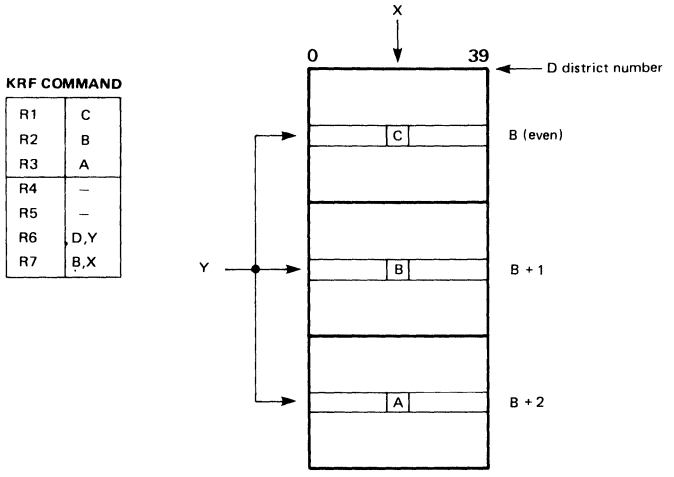
FIGURE 7 – COLORING WITH QUADRICHROME CHARACTERS



Handling long codes

The KRF command allows an easy X, Y random access or an X sequential access to/from the microprocessor from/to a memory row buffer (Figure 8).

FIGURE 8 – FIXED LONG CODES IN MEMORY 120 BYTE ROW BUFFER



VARIABLE CODES

In many cases, successive characters on screen belong to the same character set and have the same attributes. Variable codes achieve memory saving by storing B and A bytes only when it is required by exploiting the C7 bit.

C7 = 1 This is a long 3-byte code.
Character set and attribute values are completely redefined by B and A bytes.

C7 = 0 This is a short 1-byte code.
Character set and attributes value are identical to the previous code.

A further saving comes from the fact that an accentuated alphabetic character is, more often than not, followed by a not accentuated alphabetic character.
So, G_{20} or G_{21} sets are processed as one-shot escape with return to G_0. For normal operation, variable codes should obey the following rules :

- the first character code of any row (X = 0) should be long,
- a character code may be short when it has the same attributes as the previous character code and belongs to the same set.

However :

- any code belonging to G_{20} or G_{21} must be long and must be repeated if the character is double width,
- a code belonging to G_0 following a G_{20} or G_{21} code may be short.

Handling the variable codes

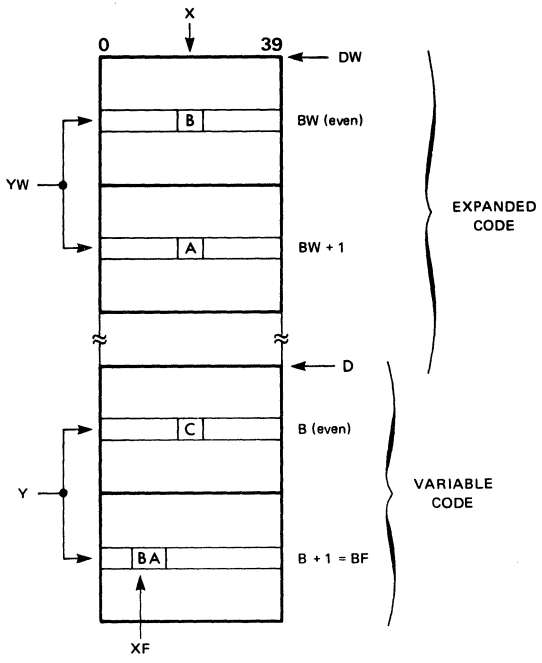
During the display process, a row of variable code should be laid in an 80/120 byte row buffer. The first buffer holds the C bytes. The second buffer holds the B, A file providing up to 20 long codes per row. (Figure 10). In the exceptionnal case when this is not enough, the second buffer overflows in the third one. Every code may then be long. Variable codes can almost always achieve a memory saving over long fixed codes and can never be worse.

The KRV command gives a very easy sequential access to/from a row buffer from/to the microprocessor. This command automatically updates the C byte and the B, A file pointers (the last one when C7 is set).

FIGURE 9 – EXPANSION/COMPRESSION MOVE

EXP and CMP commands

R1	–
R2	–
R3	–
R4	ZW, YW
R5	BF, XF
R6	D, Y
R7	B, X



Random access to a variable code is obviously not as easy. The EXP, KRE are CMP commands are designed to facilitate this task (Figure 9).

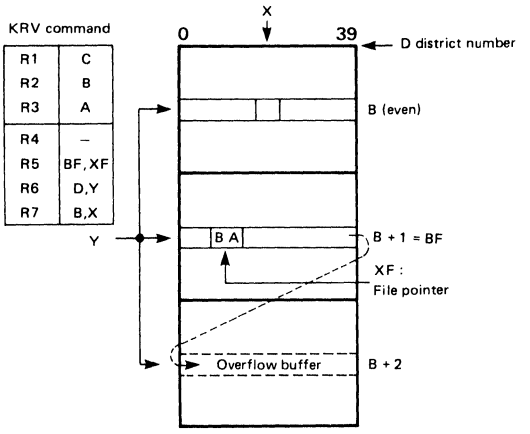
The EXP command translates a full row of variable codes into a row of expanded codes. Expanded codes are generally not displayable but very similar to the long codes.

KRE gives a random access to an expanded code and makes it appear as a regular long code.

The CMP command translates a full row of expanded code into a row of variable codes and minimizes the file size in the process.

These commands use a buffer pair as working area.

FIGURE 10 – VARIABLE CODES IN MEMORY



FIXED SHORT CODES

These fixed 16-bit codes are compatible with EF9340/41 display controllers. They achieve memory saving by another way. They may be easier to handle than variable codes. The penalty is in lesser display capabilities :

- accentuated character sets are no longer available : accentuated characters must be individually provided by the character generators,
- G'_{11} and quadrichrome sets cannot be reached,
- some attributes are latched and can be changed only while displaying a space (delimitor code).

The KRG command allows an easy access from/to an 80-byte row buffer in memory to/from the microprocessor (Figure 11). Figure 12 gives the fixed short to fixed long translation process which occurs for each row - while loading the internal row buffer - before display.

FIGURE 11 – FIXED SHORT CODES IN MEMORY 80 BYTE ROW BUFFER

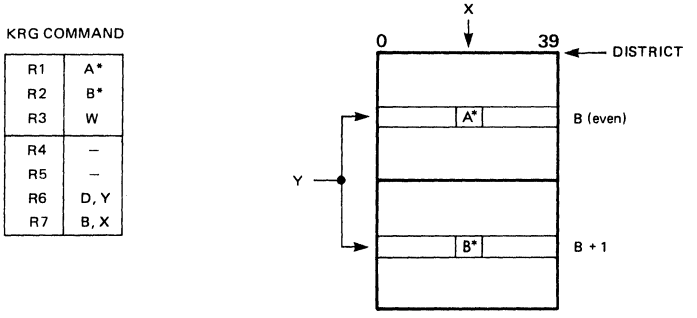
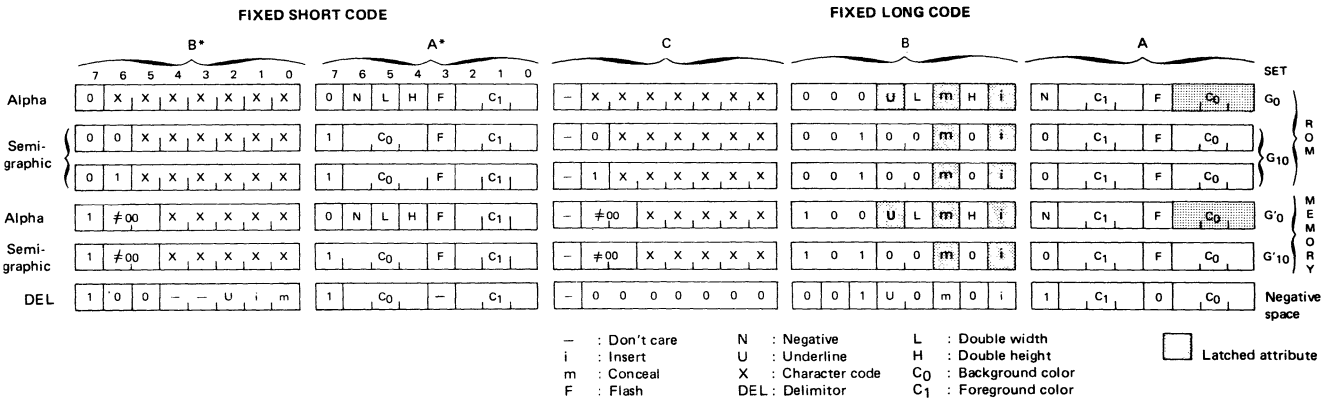


FIGURE 12 – FIXED SHORT CODE TO FIXED LONG CODE TRANSLATION



NOTES

1/ Translation process

The translation process operates through 3 elementary operations :

- Field-to-field : a character code or an attribute value (i.e : C_0, flashing) is directly loaded from short to long code.
- Field-to-constant the decoding of a short code forces the value of the equivalent long code attribute. For example, semigraphic short characters forces normal size (H = 0, L = 0) attributes.
- Latched attributes : at the beginning of each row, these attributes are reset (no underline, not concealed, no insert, black background). Then, they keep their current value until modified by either a field to field or field to constant operation.

2/ EF9340/41 compatibility

It is binary code compatible with few exceptions :

- flashing attribute is negated,
- A7 is negated in delimitors.

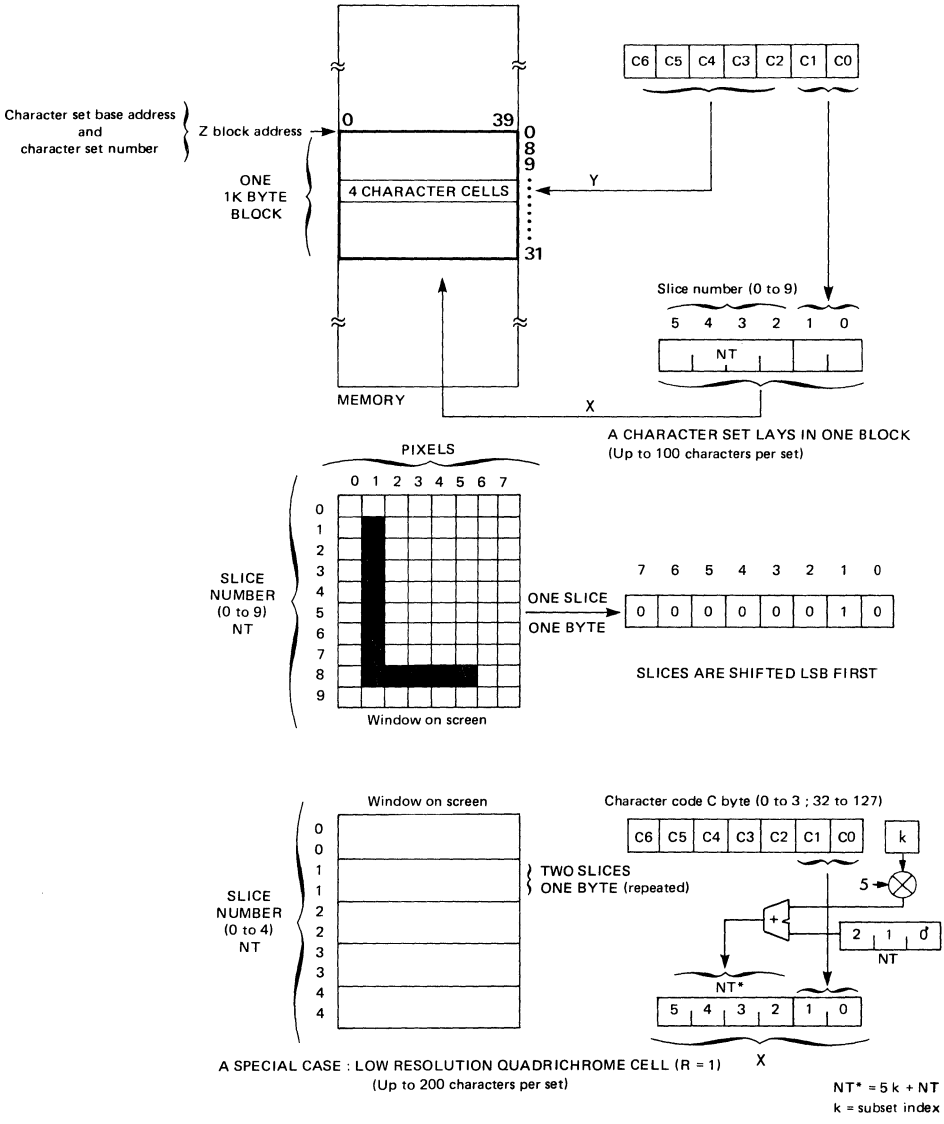
It is also display compatible with 2 exceptions concerning the underlining :

- an alphanumeric belonging to G'_0 may be underlined,
- any alphanumeric following a semigraphic cannot be underlined.

USER DEFINED CHARACTER GENERATOR IN MEMORY : DOR REGISTER

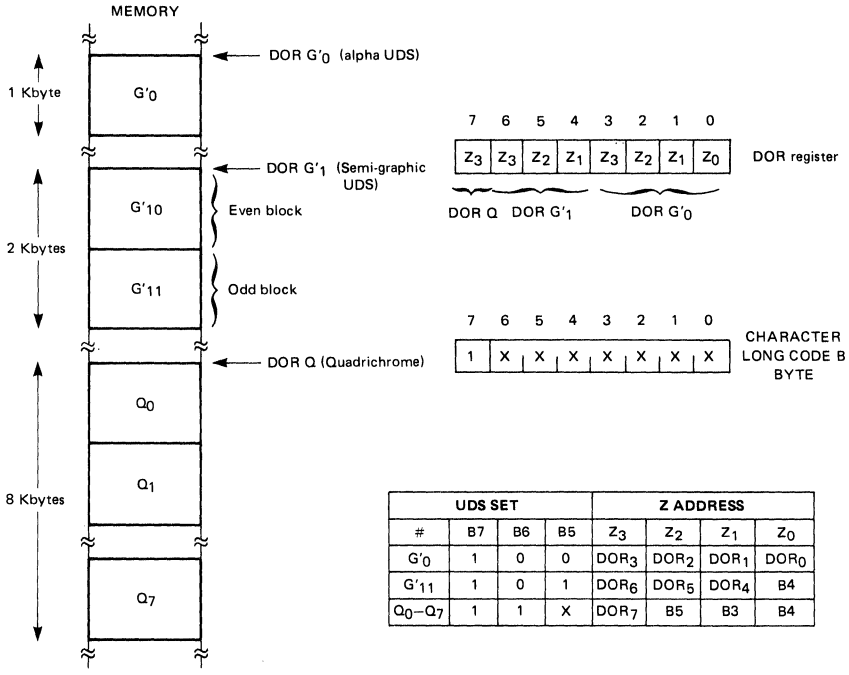
With 40 char./row, the elementary window dimensions on the screen are 10 slices x 8 pixels. Thus, a character cell holds 10 bytes in memory and 4 character cells are packed in one 40-byte buffer (Figure 13). However, 5 bytes of a low resolution quadrichrome cell are enough to fill up the window. In this case, 8 character cells can be packed in one 40-byte buffer.

FIGURE 13 – PACKING UDS CELLS IN MEMORY



The cells of one given character set should be layed in one block.
Up to 100 character cells may be addressed in each set (or 200 for low resolution quadrichrome only). The location in memory, where to fetch the sets in use, are declared by DOR register (Figure 14). For each type of set, it gives the MSB(s) of the Z block address. EF9345 reads the Z LSB(s) in the B byte of the (equivalent) long code. As usual, the character code is read in the C byte. NT is derived from the TV line rank in the row and the double height status.

FIGURE 14 – UDS FETCH TO DISPLAY



UDS SET				Z ADDRESS			
#	B7	B6	B5	Z_3	Z_2	Z_1	Z_0
G'_0	1	0	0	DOR_3	DOR_2	DOR_1	DOR_0
G'_{11}	1	0	1	DOR_6	DOR_5	DOR_4	B4
Q_0–Q_7	1	1	X	DOR_7	B5	B3	B4

LOADING USER DEFINED CHARACTER SET

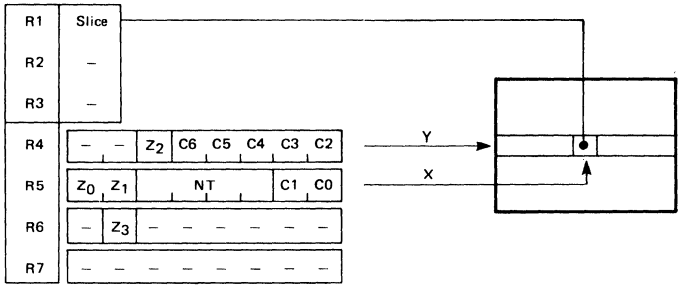
Before loading a character set into RAM, the user must :

- Assign a name to the set :
  - G'_0, G'_{10} or G'_{11} for bichrome characters.
  - From Q0 to Q7 for quadrichrome characters.
- Assign a character number to each character belonging to this set . character numbers range from 0 to 3 and 32 to 127.
  It is binary coded into 7 bits C(0:6) – C(0:6) will be loaded later on into a C byte character code in order to display the character.
- A pointer to a character slice in memory is then manufactured from :
  - the character number C(0:6)
  - the slice number NT(0:3)
  - the block number assigned to the set Z(0:3).

Figure 15 shows how to proceed with the auxiliary pointer and the OCT command.

Note :
The main pointer may be also used. When sequentially accessing slices of a given character, auto incrementation is helpless.

FIGURE 15 – ACCESSING A CHARACTER SLICE IN MEMORY USING OCT COMMAND WITH AUXILIARY POINTER



ON-CHIP CHARACTER GENERATOR

- G_0 set is common to 40 and 80 char./row modes (Figure 16 and Figure 25).
- G_{10} is the standard mosaïc set for videotex (Figure 17).
- G_{11}, G_{20} and G_{21} cannot be reached from the 16-bit short fixed codes (Figure 18 and Figure 19).

DISPLAYING THE ATTRIBUTES

1. For normal operation, a double height and/or double width character must be repeated in memory in two successive Y and/or X positions. The user may otherwise freely mix any character size.
2. The attributes are logically processed in the following order :
   - Underline or underline cursor : foreground forced on the last slice (NT = 9).
   - Flash : background periodically forced on the whole window ($\simeq$ 0.5 Hz). The phase depends on the negative attribute.
   - Conceal : background forced permanently on the whole window. A concealed character neither blinks nor is underlined.
   - Negative : exchange the background and foreground color values when set.
   - Coloring.
   - Complemented cursor mode.
   - Insert : black color forced when required.
3. Basic pixel shift frequency : f_{CLK} x 2/3 = 8 MHz.

FIGURE 16 — G_0 ALPHANUMERIC CHARACTER SET IN 40 CHARACTER/ROW MODE — EF9345

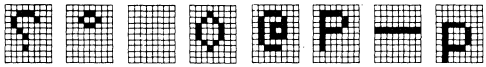
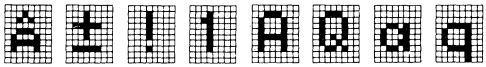
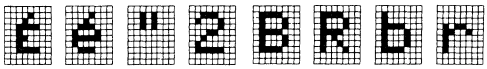
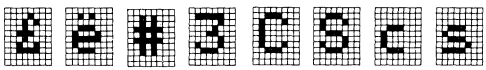
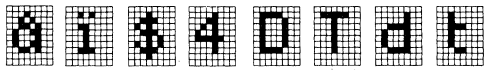
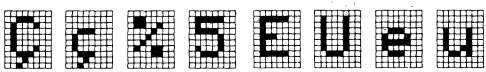
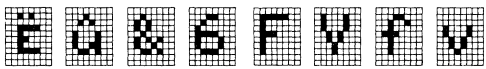
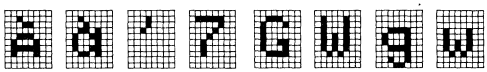
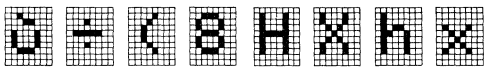
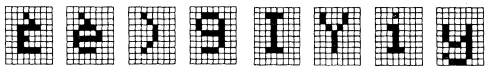
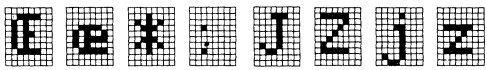
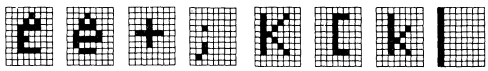
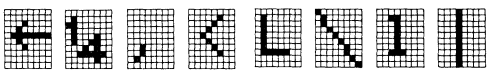
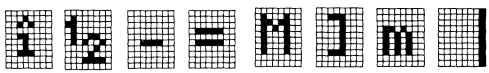
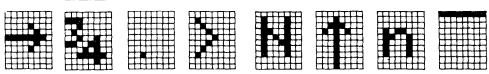
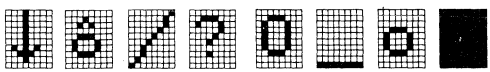
C3	C2	C1	C0	C6 = 0 0 0 0 1 1 1 1 / C5 = 0 0 1 1 0 0 1 1 / C4 = 0 1 0 1 0 1 0 1
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

FIGURE 16 bis — G_0 ALPHANUMERIC CHARACTER SET IN 40 CHARACTER/ROW MODE — EF9345 R003

C6	0	0	0	0	1	1	1	1
C5	0	0	1	1	0	0	1	1
C4	0	1	0	1	0	1	0	1

C3	C2	C1	C0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

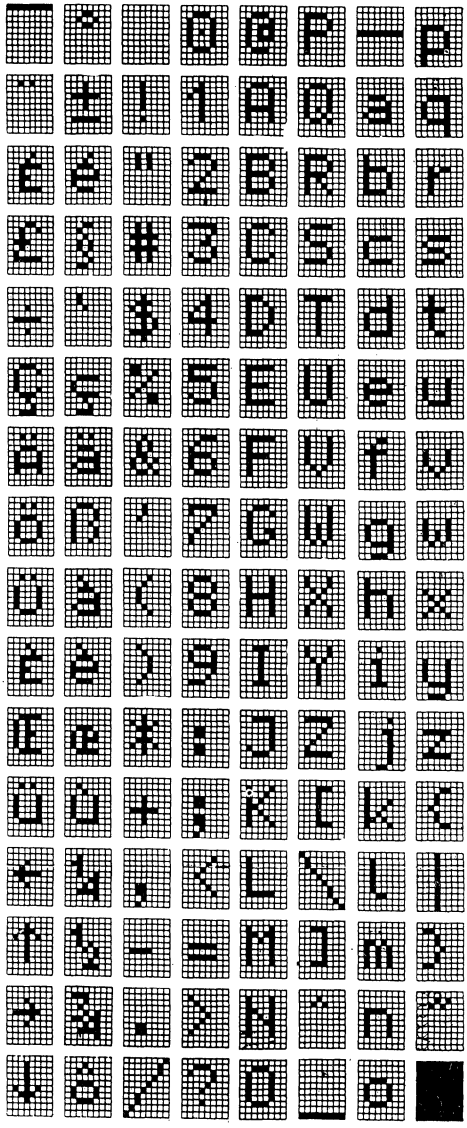


FIGURE 16 ter — G_0 ALPHANUMERIC CHARACTER SET IN 40 CHARACTER/ROW MODE — EF9345 R005

C6	0	0	0	0	1	1	1	1
C5	0	0	1	1	0	0	1	1
C4	0	1	0	1	0	1	0	1

C3	C2	C1	C0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

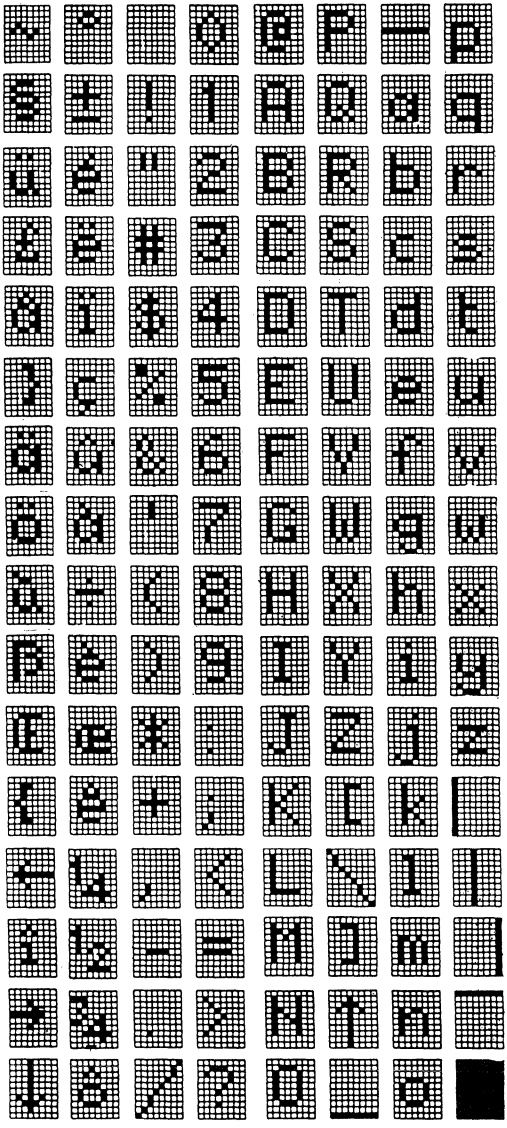


FIGURE 17 – G_{10} SEMIGRAPHIC CHARACTER SET

MOSAIC Semi-graphic SEPARATED Semi-graphic

C6	1	1	1	1	0	0	0	0
C5	0	0	1	1	0	0	1	1
C4	0	1	0	1	0	1	0	1

C3	C2	C1	C0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

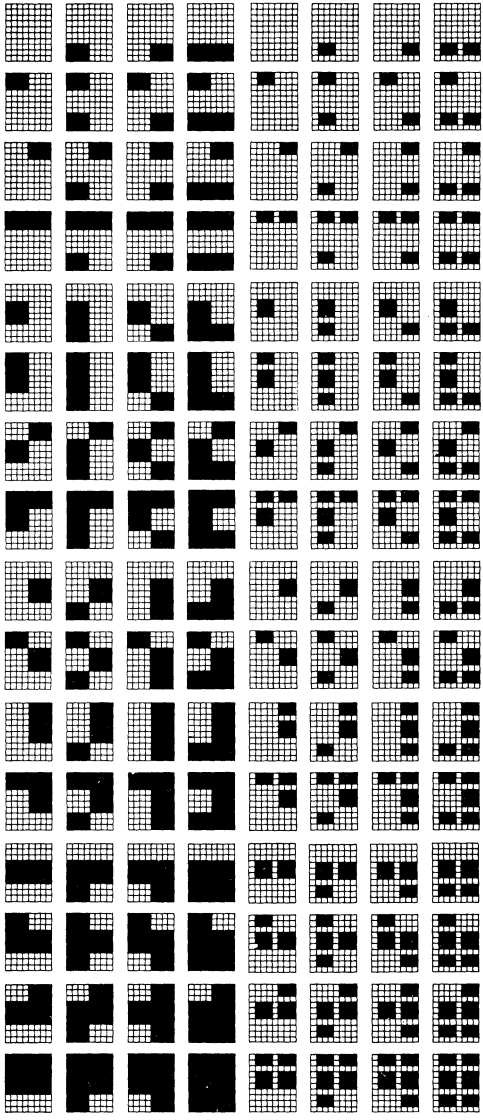


FIGURE 18 – G_{11} STROKE SET

				C5	0	0
				C4	0	1
C3	C2	C1	C0			
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

FIGURE 19 – G_{20} and G_{21} ACCENTUED CHARACTER SETS FOR 9345

	C6	0	0	1	1
	C5	0	1	0	1
B5					
0					
1					

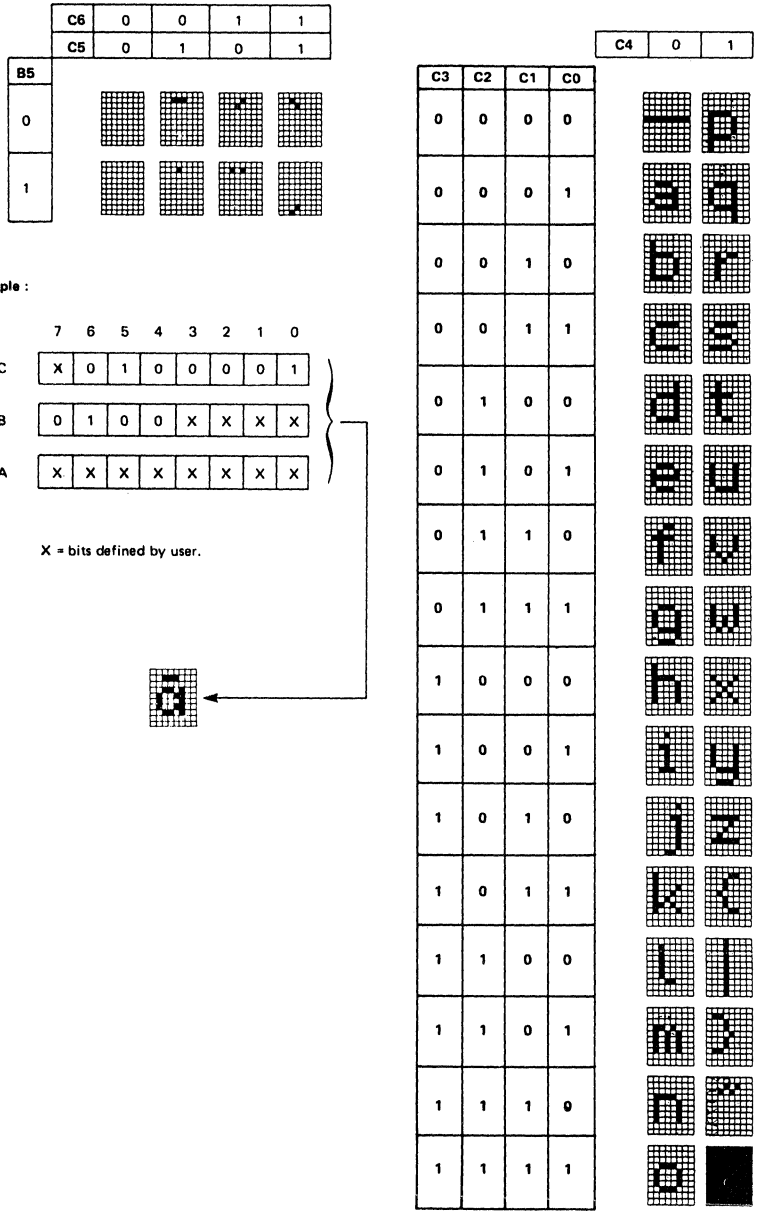
Example :

	7	6	5	4	3	2	1	0
Byte C	X	0	1	0	0	0	0	1
Byte B	0	1	0	0	X	X	X	X
Byte A	X	X	X	X	X	X	X	X

X = bits defined by user.

C3	C2	C1	C0	C4 = 0	C4 = 1
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

FIGURE 19 bis – G_{20} and G_{21} ACCENTUED CHARACTER SETS FOR 9345 - R003



Example :

	7	6	5	4	3	2	1	0
Byte C	X	0	1	0	0	0	0	1
Byte B	0	1	0	0	X	X	X	X
Byte A	X	X	X	X	X	X	X	X

X = bits defined by user.

FIGURE 19 ter – G_{20} and G_{21} ACCENTUED CHARACTER SETS FOR 9345 - R005

	C6	0	0	1	1
	C5	0	1	0	1
B5					
0					
1					

Example :

	7	6	5	4	3	2	1	0
Byte C	X	0	1	0	0	0	0	1
Byte B	0	1	0	0	X	X	X	X
Byte A	X	X	X	X	X	X	X	X

X = bits defined by user.

				C4	0	1
C3	C2	C1	C0			
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

80 CHAR/ROW CHARACTER CODES

To display pages in 80 character per row format, one of two character code formats must be selected :

- Long (12 bits) code : 4 parallel attributes and large on-chip 1024 semigraphic character set,
- Short (8 bits) code : no attribute, no semigraphic set.

Both formats address the on-chip G_0 set (128 characters 6 x 10). None allows UDS addressing.

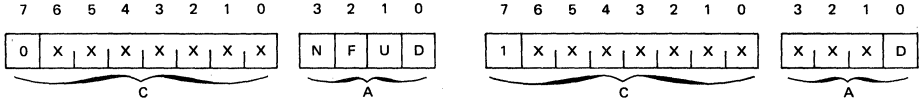
LONG CODES

Each 6 pixels x 10 lines character window on the screen is associated with a 12-bit code in memory, namely a C byte and an attribute nibble A (Figure 10). C7 bit designates the set.

- Alphanumeric set : C7 = 0.
  C(0:6) designates one out of 128 alphanumeric characters in the G_0 on-chip set. This set is common to the 40 char/row format, with the 2 right most columns truncated (see Figure 25).
  A(0:3) gives 4 parallel attributes.
- Mosaïc set : C7 = 1.
  A(1:3) and C(0:6) address a dedicated mosaïc character. Each of these address bits controls the foreground/background status of a 3 pixels x 2 lines sub-window : foreground when the bit is set.

A0 provides a color select attribute.

FIGURE 20 – 80 CHAR/ROW CHARACTER CODE



ALPHANUMERIC CHAR CODE

N = Negative
F = Flash
U = Underline
D = Color set

128 ALPHANUMERICS
In G_0 set.

MOSAIC CHAR CODE

	3 pels	3 pels	
0 1	C0	C1	DEDICATED MOSAIC SET
2 3	C2	C3	
4 5	C4	C5	
6 7	C6	A1	
8 9	A2	A3	

SHORT CODES

They are derived from the long code by giving a 0 implicit value to each bit of the A nibble : positive, not underlined, not flashing.

PACKING THE CODES IN MEMORY

Long codes are paired. A pair is packed in a 3-byte word. Therefore, the 80 codes of a row fill a 120-byte row buffer (Figure 21). The left most position on the screen is even. Its corresponding C byte is at the beginning of the first buffer. The next position on the screen is odd. Its corresponding C byte is at the beginning of the second buffer. Both nibbles are packed in the third buffer. With short codes, the same scheme yields 80-byte row buffers.

ACCESS TO THE CODES IN MEMORY

KRL command transfers 12 bits from/to the R1 and R3 registers to/from memory. The read modify write operation, necessary to write the A nibble in memory, is automatically performed provided that the A nibble is repeated in the R3 register (Figure 22). Dedicated auto-incrementation is also performed when required.

KRC command does a similar job for the short codes (Figure 23).

A very simple scheme allows the microprocessor to transcode an horizontal screen location into a pointer (Figure 24). The joint use of this scheme with the dedicated command alleviates all the packing/unpacking troubles.

FIGURE 21 – 80 CHAR/ROW CODE PACKING

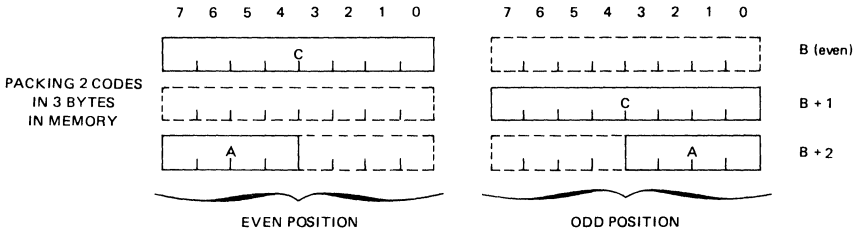


FIGURE 22 – KRL COMMAND : SEQUENTIAL ACCESS TO LONG CODES

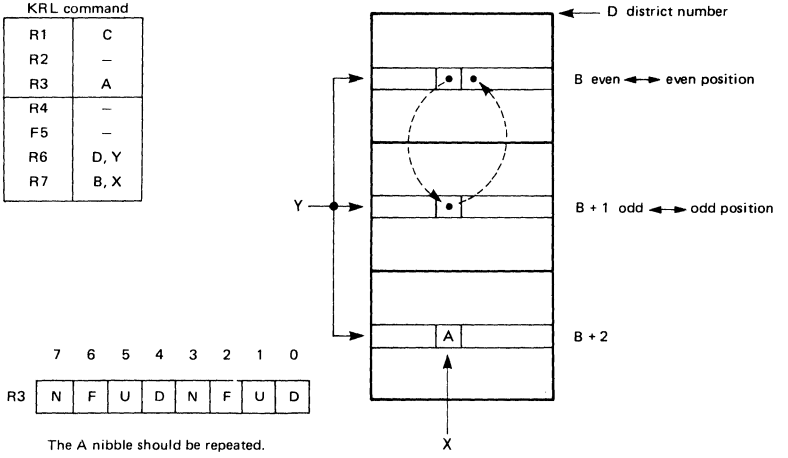


FIGURE 23 – KRC COMMAND SEQUENTIAL ACCESS TO SHORT CODES

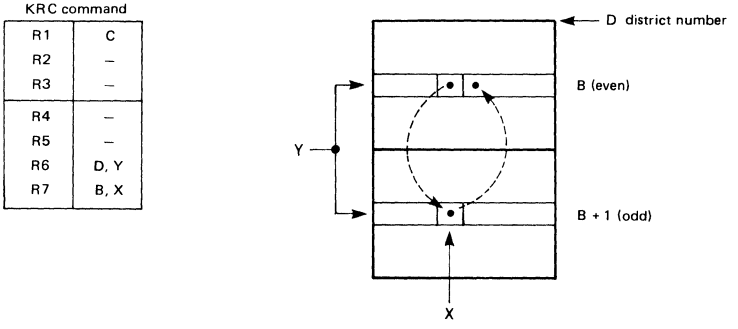
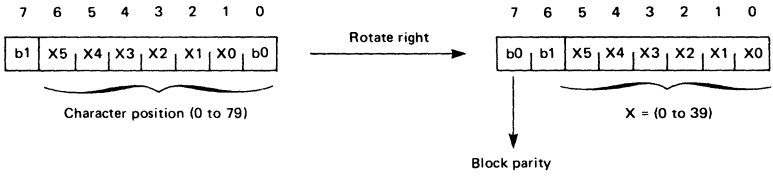


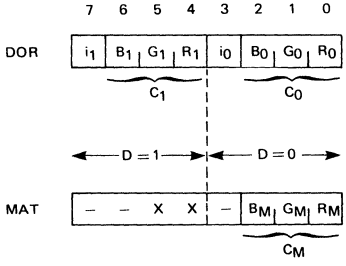
FIGURE 24 – TRANSCODING AN HORIZONTAL SCREEN LOCATION INTO A R7 POINTER



DISPLAYING THE ATTRIBUTES - DOR REGISTER

Short code and mosaïc characters are not flashing, not underlined and "positive".
The attributes are processed in the following order :

- Underline or underlined cursor : foreground is forced on the last slice (NT = 9).
- Flash : background is periodically (0.5 Hz – 50 %) forced on all the window. The phase depends on the negative attribute.
- Color select : a "positive" character is displayed with a background color same as the margin color. The foreground color is selected in DOR register by the D attribute.
- Negative : when the character is negative, background and foreground colors are exchanged. In complemented CURSOR position, these colors are complemented.
- Insert : the D attribute selects one insert value in DOR register. This attribute is then processed up to the current insertion mode (see screen format and attribute insert section).



The pixel shift frequency is f_{CLK} (12 MHz).

D	N	BACKGND COLOR	FOREGND COLOR	i
0	0	C_M	C_0	i0
0	1	C_0	C_M	i0
1	0	C_M	C_1	i1
1	1	C_1	C_M	i1

FIGURE 25 — G_0 ALPHANUMERIC CHARACTER SET IN 80 CHARACTER/ROW MODE — EF9345

C7	0	0	0	0	0	0	0	0
C6	0	0	0	0	1	1	1	1
C5	0	0	1	1	0	0	1	1
C4	0	1	0	1	0	1	0	1

C3	C2	C1	C0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

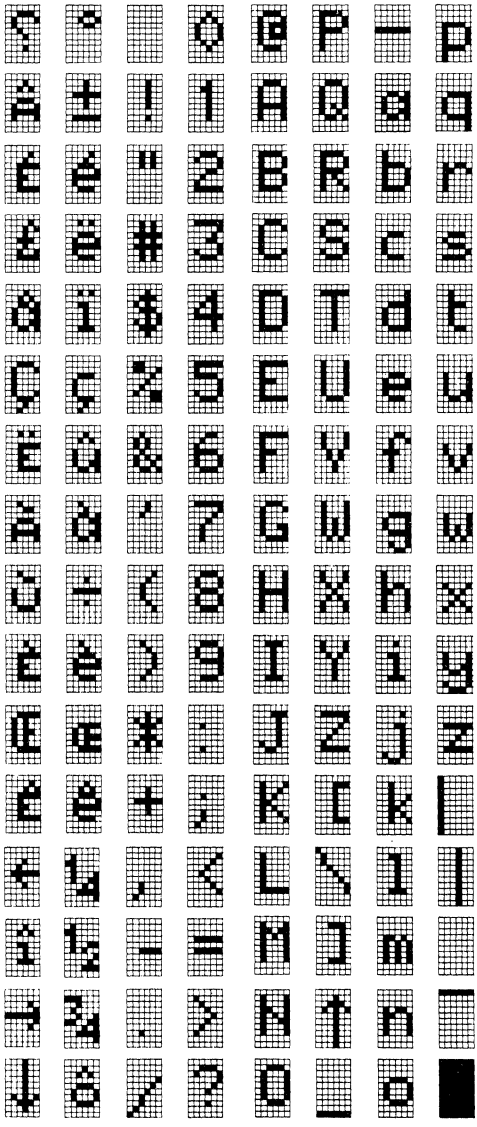


FIGURE 25 bis — G_0 ALPHANUMERIC CHARACTER SET IN 80 CHARACTER/ROW MODE — EF9345 R003

C7	0	0	0	0	0	0	0	0
C6	0	0	0	0	1	1	1	1
C5	0	0	1	1	0	0	1	1
C4	0	1	0	1	0	1	0	1

C3	C2	C1	C0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

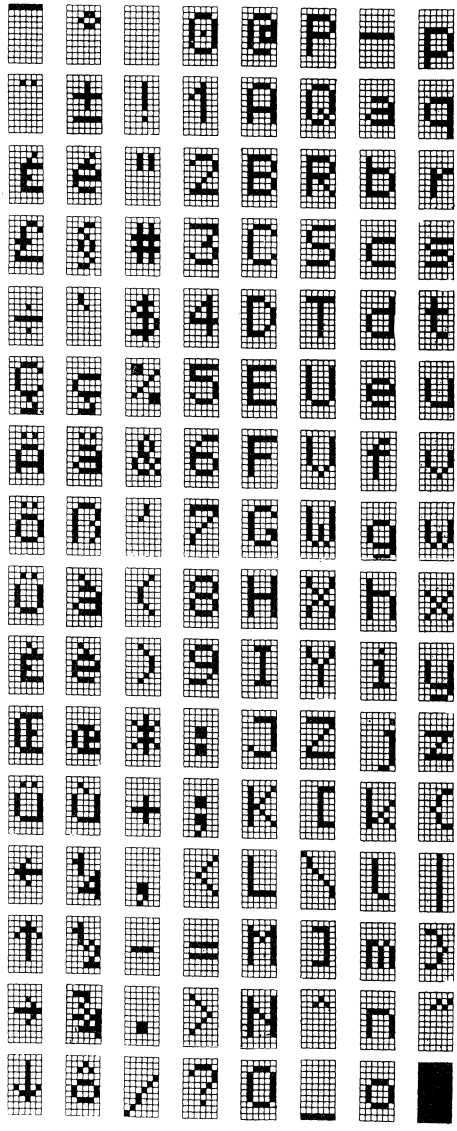
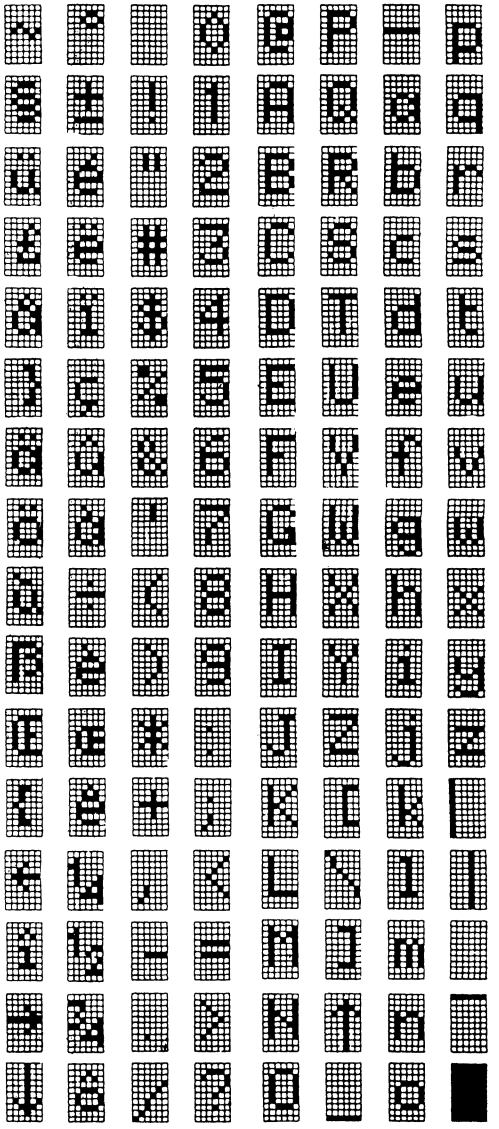


FIGURE 25 ter — G_0 ALPHANUMERIC CHARACTER SET IN 80 CHARACTER/ROW MODE — EF9345 R005

C7	0	0	0	0	0	0	0	0
C6	0	0	0	0	1	1	1	1
C5	0	0	1	1	0	0	1	1
C4	0	1	0	1	0	1	0	1

C3	C2	C1	C0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1



MICROPROCESSOR ACCESS COMMANDS

A microprocessor bus cycle may transfer one byte from/to the microprocessor to/from a directly addressable register. These registers provide an indirect access :

- to/from 5 on-chip indirect registers : ROR, DOR, MAT, PAT and TGS.
- to/from the private memory.

Due to address/data multiplexing, a bus cycle is a 2 phase process (see Timing diagram 1 or Timing diagram 2).

- A8 = $\overline{LCS}$
  This is the latched value of $\overline{CS}$ input pin.

EF9345 is selected when the following condition is met : ASN = 2 (Hexa) and $\overline{LCS}$ = 0.

Therefore, EF9345 is mapped in the hexadecimal microprocessor addressing space from XX20 to XX2F, where XX is up to the user. When EF9345 is not selected, its AD bus pins float and no register can be modified.

ADDRESS PHASE

The falling edge of AS latches to AD(0:7) bus state and $\overline{CS}$ signal into the temporary A address register (Figure 26).

- A(0:2) = i
  This register index designates one out of 8 direct access registers R_i.
- A3 = XQR
  This is the execution request bit.
- A(4:7) = ASN
  This is the Auto-Selection Nibble.

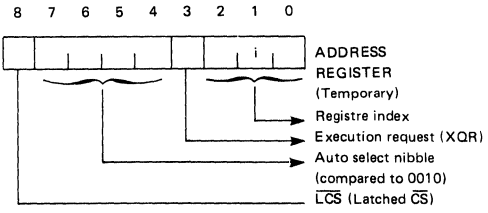
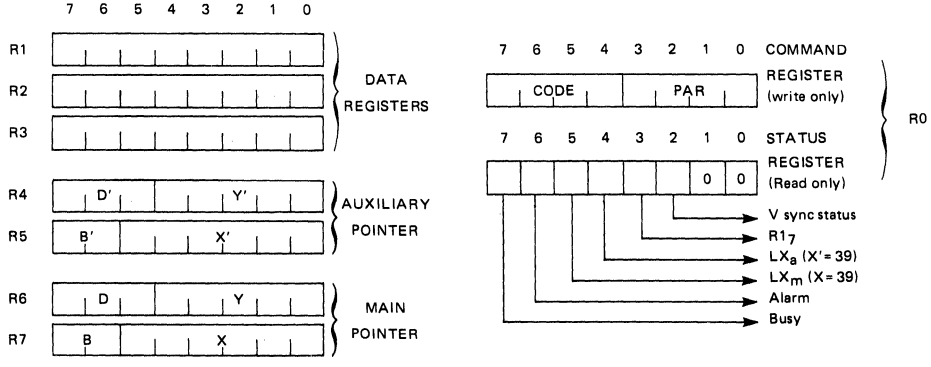


FIGURE 26 – DIRECT ACCESS REGISTERS



DATA PHASE - REGISTERS

When EF9345 is selected and while AS input is low, the R_i register is accessed.
R0 designates a write-only COMMAND register or a read-only STATUS register.
R1 to R7 hold the arguments of a command. They are read/write registers.
R1, R2, R3 are used to transfer the data.
R4, R5 hold the Auxiliary Pointer (AP).
R6, R7 hold the Main Pointer (MP).
(See memory organization ; Pointer section for pointer structure).

COMMAND REGISTER

This register holds a 4-bit command type and 4 bits of orthogonal parameters (see COMMAND TABLE).

Type

There are 4 groups of command :

- The IND command which gives access to on-chip resources,
- The fixed format character code transfer commands,
- The variable character code handling commands,
- The general purpose commands.

Parameters

R/$\overline{W}$:Direction
1 : to DATA registers (R1, R2, R3)
0 : from DATA registers.

r : Internal ressource index (see Figure 27)

I : Auto-incrementation
1 : with post auto-incrementation
0 : without auto-incrementation.

p : Pointer select
1 : auxiliary pointer
0 : main pointer

s, $\overline{s}$: Source, destination select
01 : source : MP ; destination : AP
10 : source : AP ; destination : MP

$\overline{a}$, a : Stop condition
01 : stop at end of buffer
10 : no stop.

STATUS REGISTER

This is a read-only, direct access register.

S7 : BUSY — BUSY is set at the beginning of any command execution. It is reset at completion.

S6 : AI, S5 : LX_m, S4 : LX_a — LX_m or LX_a is set when respectively the main pointer or the auxiliary pointer holds X = 39 before a possible incrementation.
The alarm bit S6 is set when LX_m or LX_a is set and an incrementation is performed after access.

S3 : Gives the MSB value of R_1.

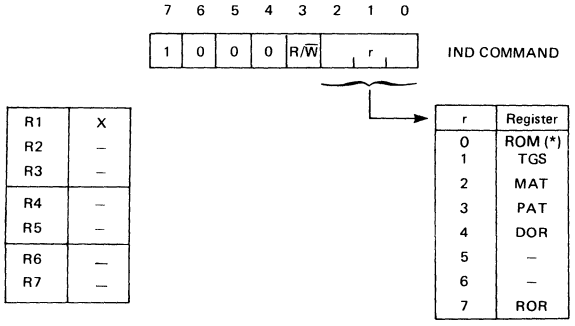
S2 : Gives the vertical synchronization signal state.
This is maskable by the VRM command.

S1 = S0 = 0 Not used.

S3 to S6 are reset at the beginning of any command.

The COMMAND TABLE shows every command able to set, each of these status bits, after completion.

FIGURE 27 – INDIRECT ON-CHIP RESOURCE ACCESS



* Note: A slice in 40C only can be read from the internal character generator. The slice address must be initialized in R6, R7.

NOTES ON COMMAND EXECUTION

1. The execution of any command starts at the trailing edge of DS when (and only when) :
   - EF9345 has been selected,
   - XQR has been set,
   at the previous AS falling edge.

This scheme allows loading a command and its argument in any order. For instance, a command, once loaded, may be re-executed with new or partly new arguments.

2. At power on, the busy state is undeterminated.
   It is recommanded to load first a dummy command with XQR = 1 before any effective command.

3. While Busy is set, the current command is under execution. Register access is then restricted .

   Register access with XQR = 0
   - Read STATUS is effective.
   - Write COMMAND or any other register access are ineffective.

That is to say, the microprocessor reads undetermined values and may not modify a register.

Register access with XQR = 1
- Read STATUS or write COMMAND are effective,
- Access to other registers is ineffective.

However, the previous command is aborted and the new command execution launched (with an initial state undetermined for registers and memory locations handled by the aborted command).

4. Execution suspension

The execution of any command (except VRM, VSM) is suspended during the last and first TV line of an active row. This is because the memory bus cannot be allocated for microprocessor access during this 104 μs period.
This holds too for internal resource access because on-chip data transfer uses internal data memory bus.

IND COMMAND (See figure 27)

This command transfers one byte between R1 and an internal resource. The r parameter designates one on-chip indirect register.

FIXED FORMAT CHARACTER CODE ACCESS : KRF, KRG, KRL KRC

Each of these commands is dedicated to transfer one complete character code between DATA registers and memory. MP is exclusively used.

KRF transfers 24 bits.
KRG transfers 16 bits.
KRL transfers 12 bits.
KRE transfers 8 bits.

Code packing, pointer and data structures are explained in the corresponding character code section.

When auto-incrementation is enabled, MP is automatically updated after access so as to point to the next location. This location corresponds to the next right position on screen. When last position (X = 39) is accessed, LX_m is set. When last position is accessed with auto-incrementation, alarm is also set. MP is then pointing back at the beginning of the row : there is no automatic Y incrementation.

VARIABLE CODE HANDLING COMMANDS : KRV EXP, CMP, KRE

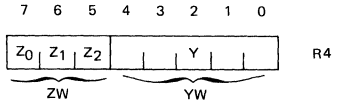
An overview on these commands is given in "handling the variable codes" (40 char./row section).

KRV uses R5 to point the attribute file. LX_a is set when this file is full (the last attribute pair has been accessed).

EXP and CMP use MP and R5 in the same way as KRV. Furthermore, R4 points to a working double buffer. These two commands process a whole row buffer and stop either at the end of the row buffer or when the file overflows. In the last case, the alarm bit is set.

KRE uses MP to point to a buffer and R4 to point to a working double buffer. R5 is unused. In other respects, KRE is identical to KRL.

For these commands, R4(5:7) hold the LSB's block address of the working buffer W.



ZW_3 is given by bit 6 of R6.

GENERAL PURPOSE ACCESS TO A BYTE - OCT

This command uses either MP or AP pointer.
When MP is in use, an overflow yields to a Y incrementation.

MOVE BUFFER COMMANDS : MVB, MVD, MVT

These are memory to memory commands which use R1 as working register.
MVB transfers a byte from source to destination, post-increments the 2 pointers and iterates until the stop condition is met. MVD and MVT are similar but work respectively with 2 byte word and 3 byte word. That is to say, MVB works on buffers, MVD on double buffers and MVT on triple buffers. If the parameter a = 1, the process stops when either source or destination buffer end is reached. If the parameter a = 0, the process never stops until aborted. In this case, main pointer overflow yields to a Y incrementation in MP. So, a whole block or page may be initialized.

MISCELLANEOUS COMMANDS : INY, VRM and VSM

INY command increments Y in MP.

VRM and VSM respectively reset and set a vertical synchronization status mask. When the mask is set, status bit S2 remains at 0. When the mask is reset, status S2 follows the vertical sync. state : it is reset for 2 TV lines per frame and stays at 1 during the remaining period. It becomes readable by the microprocessor from the status register. After power on, the mask state is undetermined.

COMMAND TABLE

TYPE	MEMO	CODE				PARAMETER				STATUS				ARGUMENTS							EXECUTION TIME (1)	
		7	6	5	4	3	2	1	0	AI	LX_m	LX_a	$R1_7$	R1	R2	R3	R4	R5	R6	R7	WRITE	READ
INDIRECT	IND	1	0	0	0	R/$\overline{W}$		r		0	0	0	0	D	–	–	–	–	MP		2	3.5
40 CHARACTERS - 24 BITS	KRF	0	0	0	0	R/$\overline{W}$	0	0	I	X	X	0	0	C	B	A	–	–	MP		4	7.5
40 CHARACTERS - 16 BITS	KRG	0	0	0	0	R/$\overline{W}$	0	1	I	X	X	0	0	A*	B*	W	–	–	MP		5.5	7.5
80 CHARACTERS - 8 BITS	KRC	0	1	0	0	R/$\overline{W}$	0	0	I	X	X	0	0	C	–	–	–	–	MP		9	9.5
80 CHARACTERS - 12 BITS	KRL	0	1	0	1	R/$\overline{W}$	0	0	I	X	X	0	0	C	–	A	–	–	MP		12.5	11.5
40 CHARACTERS VARIABLE	KRV	0	0	1	0	R/$\overline{W}$	0	0	I	X	X	X	X	C	B	A	–	XF	MP		(2) 3 + 3 + j	3.5 + 6 • j
EXPANSION	EXP	0	1	1	0	0	0	0	0	X	0	X	0	C	B	A	PW	XF	MP		(3) < 247	–
COMPRESSION	CMP	0	1	1	1	0	0	0	0	X	0	X	0	C	B	A	PW	XF	MP		(3) < 402	–
EXPANDED CHARACTERS	KRE	0	0	0	1	R/$\overline{W}$	0	0	I	X	X	0	0	C	B	A	PW	–	MP		4	7.5
BYTE	OCT	0	0	1	1	R/$\overline{W}$	p	0	I	X	X	X	0	D	–	–	AP		MP		4	4.5
MOVE BUFFER	MVB	1	1	0	1	s	$\overline{s}$	$\overline{a}$	a	0	0	0	0	W	–	–	AP		MP		(2) 2 + 4.n	–
MOVE DOUBLE BUFFER	MVD	1	1	1	0	s	$\overline{s}$	$\overline{a}$	a	0	0	0	0	W	–	–	AP		MP		(2) 2 + 8.n	–
MOVE TRIPLE BUFFER	MVT	1	1	1	1	s	$\overline{s}$	$\overline{a}$	a	0	0	0	0	W	–	–	AP		MP		(2) 2 + 12.n	–
CLEAR PAGE (4) - 24 BITS	CLF	0	0	0	0	0	1	0	1	X	X	0	0	C	B	A	–	–	MP		< 4700 (1 K code)	–
CLEAR PAGE (4) - 16 BITS	CLG	0	0	0	0	0	1	1	1	X	X	0	0	A*	B*	W	–	–	MP		< 5800 (1 K code)	–
VERTICAL SYNC MASK SET	VSM	1	0	0	1	1	0	0	1	0	0	0	0	–	–	–	–	–	–	–	1	–
VERTICAL SYNC MASK RESET	VRM	1	0	0	1	0	1	0	1	–	–	–	–	–	–	–	–	–	–	–	1	–
INCREMENT Y	INY	1	0	1	1	0	0	0	0	0	0	0	0	–	–	–	–	–	Y	–	2	–
NO OPERATION	NOP	1	0	0	1	0	0	0	1	–	–	–	–	–	–	–	–	–	–	–	1	–

p : Pointer select
1 : auxiliary pointer
0 : main pointer.

s, $\overline{s}$: Source, destination
01 : source = MP ; destination = AP
10 : source = AP ; destination = MP

$\overline{a}$, a : Stop condition
01 : stop at end of buffer
10 : no stop

r : Indirect register number.

– : Not affected
W : Used as working register
PW (ZW, YW) : Working buffer
X : Set or Reset
XF : X File
I : Pointer incrementation
D : Data
MP : Main pointer
AP : Auxiliary pointer.

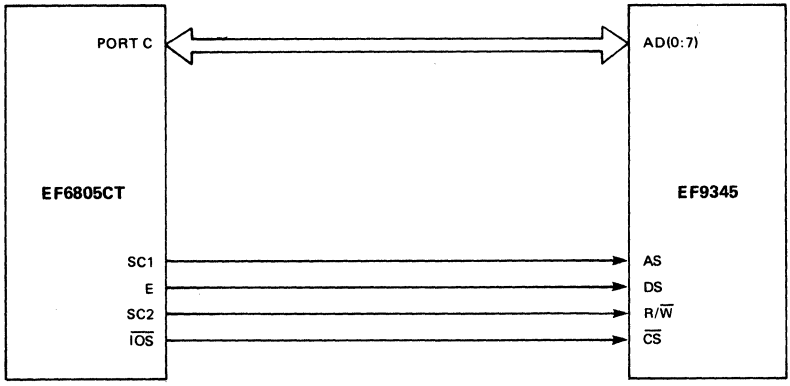
(1) Unit : 12 clock periods ($\simeq 1\ \mu s$) without possible suspension.

(2) n : total number of words $\leqslant 40$; j = 1 for long codes, j = 0 for short codes.

(3) Worst case (20 long codes + 20 short codes).

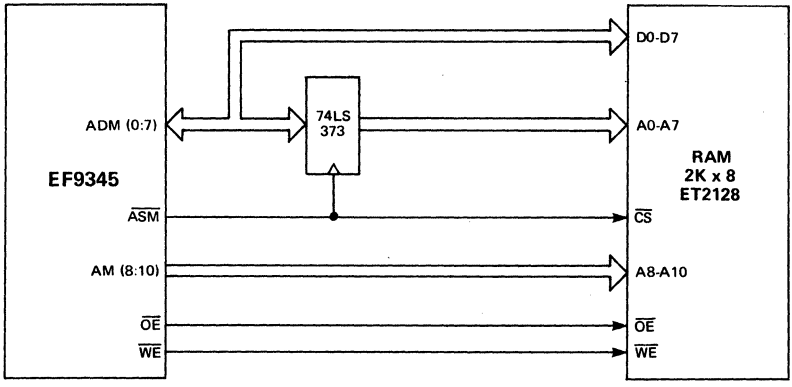
(4) These commands repeat KRF or KRG with Y incrementation when X overflows. When the last position is reached in a row, Y is incremented and the process starts again on the next row. These commands stop only with abort.

INTERFACE WITH EF6805CT



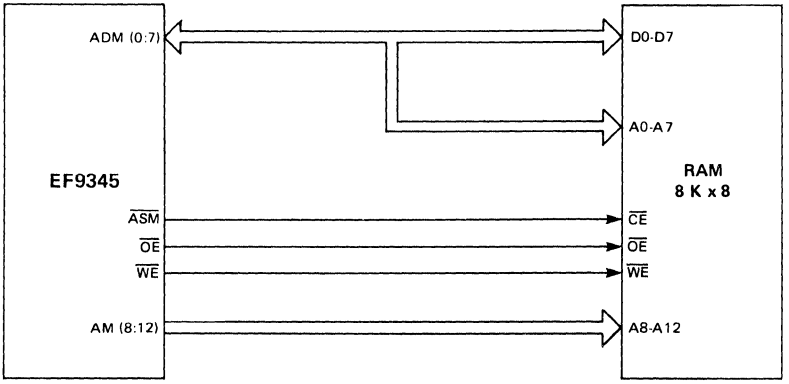
MINIMUM APPLICATION WITH 2K x 8 MEMORY

One page memory terminal in 16-bit fixed format or 24-bit compressed format.



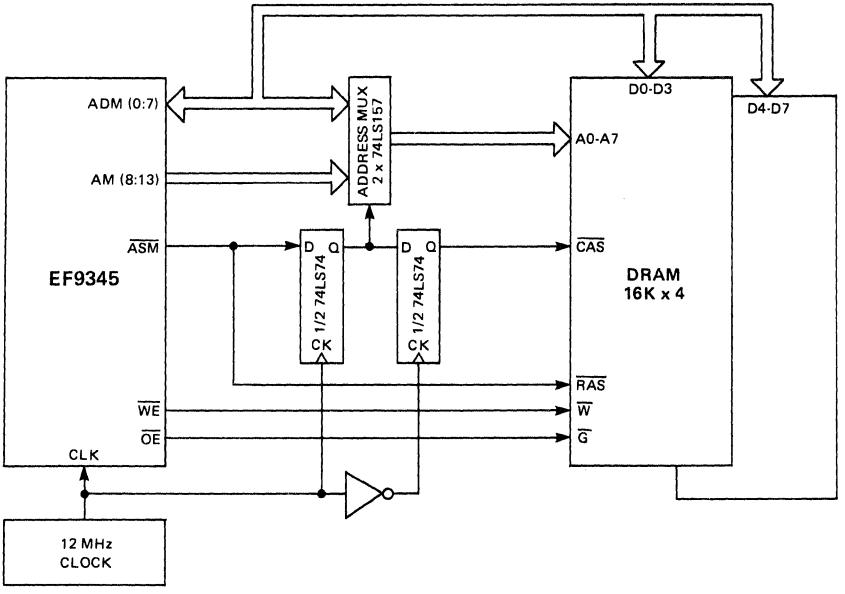
TYPICAL APPLICATION WITH 8K x 8 DYNAMIC OR PSEUDO-STATIC RAM

Multipage terminal with possibility of multiple user definable character sets.

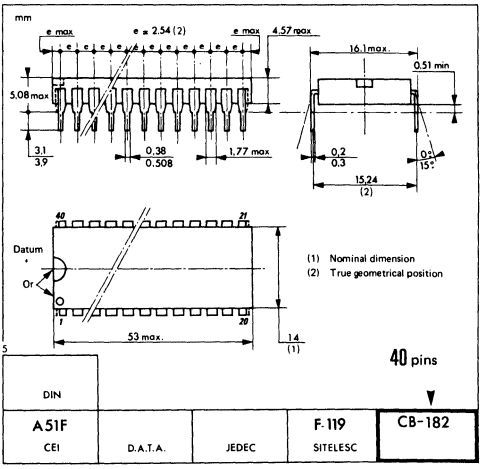


MAXIMUM APPLICATION WITH 16K x 8 MEMORY

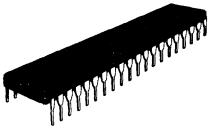
Multipage terminal with user definable character sets and buffer areas.



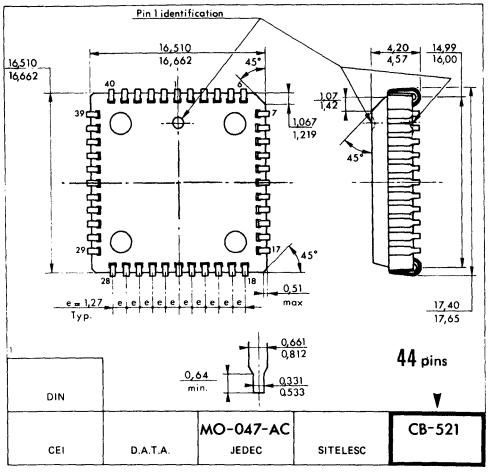
PHYSICAL DIMENSIONS



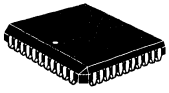
CB-182



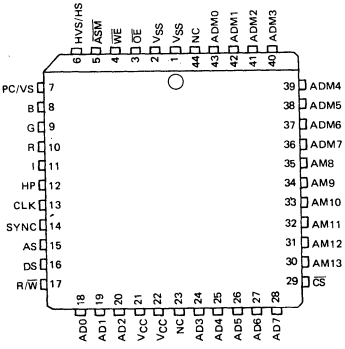
P SUFFIX
PLASTIC PACKAGE



CB-521



FN SUFFIX
SURPICOP
(PLASTIC CHIP-CARRIER)



ORDERING INFORMATION

EF9345 XX RYYY

- RYYY → Character generator
- XX → Package: P, FN

THOMSON COMPONENTS MOSTEK

EF9367

GRAPHIC DISPLAY PROCESSOR

ADVANCE INFORMATION

This GDP is a true high resolution graphic display processor, which contains all the functions required to process vector generation at a very high speed and to generate all the timing signals required for interfacing interlaced or non interlaced video data on a raster scan CRT display compatible with 525 line or the CCIR 625 line standards.

The GDP's main features are :

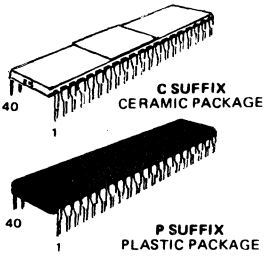
- Selectable resolutions in black and white or color :
  Vertical resolution : 525 line monitor (208 or 416). 625 line monitor (256 or 512).
  Horizontal resolution : 256, 320*, 384*, 512, 640*, 768*, 1024, full screen. (*) with external PROM.
- High speed vector plot well suited to animation - 4 types of lines.
- Multiplexed address and refresh for 16K or 64K dynamic RAMs.
- No limitation on the number of selectable memory planes (colors, grey levels or any other attributes)
- Multipage application capability
- On-chip full ASCII character generator (96) - maximum alphanumeric screen density : 170 x 57 - programmable sizes and orientations
- Direct interfacing with the monitor through the composite synchro and blanking signals
- Automatic allocation of display memory in refresh, write, dump, and display cycles
- Light pen registers and control signals
- Three types of interrupt requests
- Fully static design
- TTL compatible I/O
- Single + 5 volt supply.

MOS

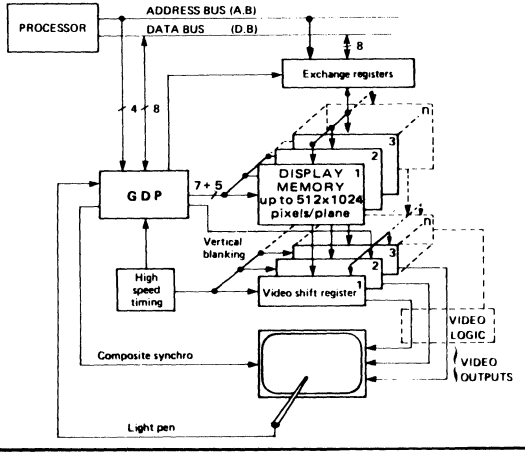
(N - CHANNEL, SILICON-GATE)

GRAPHIC DISPLAY PROCESSOR (GDP)

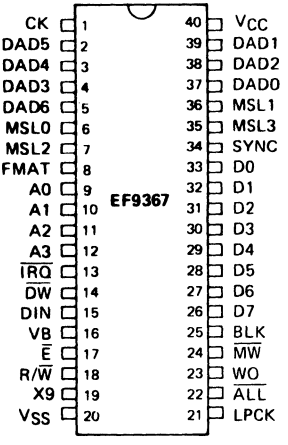
CASE CB-182



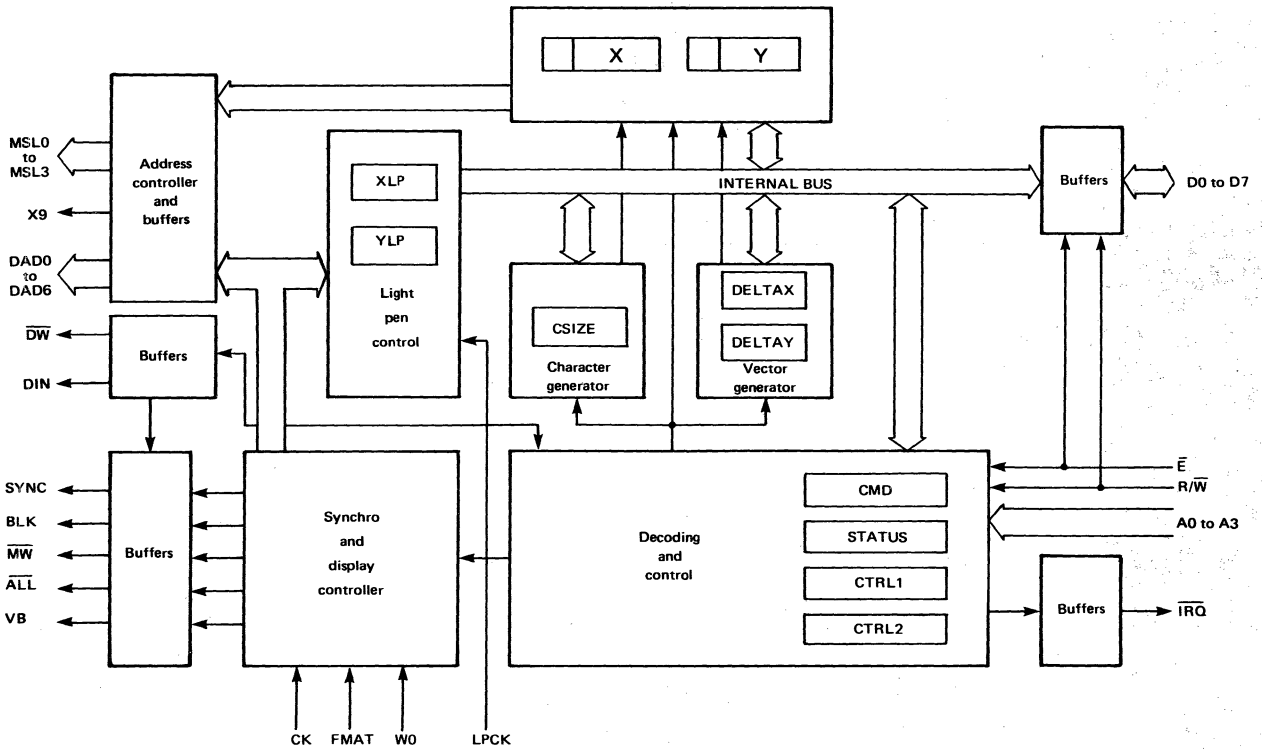
TYPICAL APPLICATION



PIN ASSIGNMENT



BLOCK DIAGRAM



GENERAL DESCRIPTION

Developed using NMOS technology, the GDP is an intelligent raster scan video display controller, fully programmable via an eight-bit microprocessor bus. Besides all the timing logic functions required to generate the video, sync and blanking signals, the GDP includes two hardwired display processors : a vector and a character generator.

This unique feature allows an ultrafast screen writing speed (the 1024 dot diagonal may be written in less than 1.4 ms) at almost no microprocessor processing cost.

The GDP is particularly well-suited to all applications in which the display memory is not directly addressed by the MPU. This feature allows a total asynchronism between the MPU and the GDP memory cycles and preserves the whole MPU memory addressing space.

Nevertheless, where direct exchange between the microprocessor and the memory is necessary, the on-chip allocation controller will allow this exchange without display interference.

The GDP is programmable using 11 internal registers occupying 16 consecutive addresses. These registers can also be modified by the GDP's hardwired processors while a command is being executed.

Note : A summary of data codes and registers is given in the **Register address table.** Hexadecimal values are subscripted 16 and the register bits are numbered as follows :

MSB	7	6	5	4	3	2	1	0	LSB

MAXIMUM RATINGS

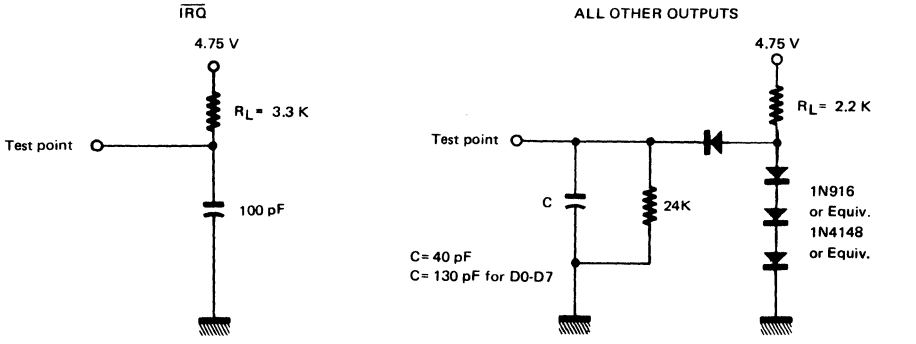
Rating	Symbol	Value	Unit
Supply voltage	V_{CC}	−0.3 to+ 7.0	V
Input voltage	V_{in}	−0.3 to+ 7.0	V
Operating temperature	T_A	0 to+ 70	°C
Storage temperature	T_{stg}	− 55 to+150	°C

The GDP inputs are protected against high static voltages and electric fields ; nevertheless, normal precautions should be taken to avoid voltages above the limit values on this high impedance circuit.

STATIC ELECTRICAL CHARACTERISTICS (V_{CC} = 5 V ± 5 %, V_{SS} = 0, T_A = 0 to 70° C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input high voltage except CK	V_{IH}	V_{SS}+ 2.2	–	V_{CC}	V
Input high voltage CK	V_{IHCK}	V_{SS} + 3.5	–	V_{CC}	V
Input low voltage	V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
Input leakage current (V_{in} = 0 to 5.25 V, V_{CC} = max)	I_{in}	–	1.0	2.5	µA
Output high voltage (I_{load} = −100 µA, V_{CC} = min)	V_{OH}	V_{SS} + 2.4	–	–	V
Output low voltage (I_{load} = 1.6 mA, V_{CC} = min)	V_{OL}	–	–	V_{SS} + 0.4	V
Supply current	I_{CC}	–	80	–	mA
Capacitance (V_{in} = 0, T_A = 25° C, f = 1.0 MHz)	C_{in}, C_{out}	–	–	12	pF

TEST LOADS

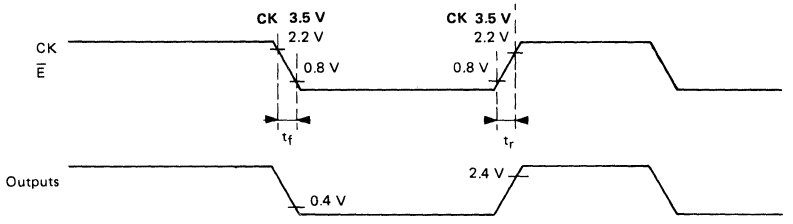


DYNAMIC OPERATING CONDITIONS

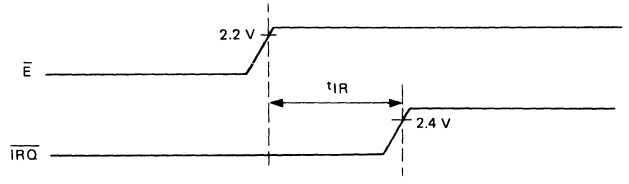
(V_{DD} = 5,0 V ± 5 %, V_{SS} = 0V ; T_A = 0 to+ 70° C unless otherwise noted)

Time (ns)	Symbol	Min	Max
Clock period	t_{CK}	560	
CK pulse width, low	t_{CKL}	330	
CK pulse width, high	t_{CKH}	190	
CK low to valid DAD	CKLDAD		320
CK high to valid DAD	CKHDAD		180
CK low to valid SYNC	CKLSYNC		300
CK low to valid BLK	CKLBLK		310
CK low to valid VB	CKLVB		500
CK low to valid $\overline{ALL}$	CKLALL		300
CK low to valid $\overline{MSL}$	CKLMSL		300
CK low to valid $\overline{DW}$	CKLDW		310
CK low to valid $\overline{MFREE}$ low	CKLMFRL		330
CK low to valid $\overline{MFREE}$ high	CKLMFRH		500
CK low to valid DIN	CKLDIN		310
CK low to valid $\overline{IRQ}$	CKLIRQ		1500
CK low to valid $\overline{WHITE}$	CKLWHI		530
$\overline{E}$ pulse width, low	t_{EL}	450	
$\overline{E}$ pulse width, high	t_{EH}	430	
Address pre-setup time	t_{AS}	160	
Address hold time	t_{AH}	10	
Data pre-setup time (write)	t_{DSW}	195	
Data setup time (read)	t_{DDR}		320
Data hold time (read)	t_{DHR}	10	
$\overline{IRQ}$ release time	t_{IR}		1600
LPCK high to $\overline{WHITE}$ high (if command 08_{16})	LPHW		1600
LPCK high to $\overline{IRQ}$ low	LPHIRQ		1600
LPCK high hold time	t_{LPCKH}	150	
CK and $\overline{E}$ rise times	t_r		20
CK and $\overline{E}$ fall times	t_f		20

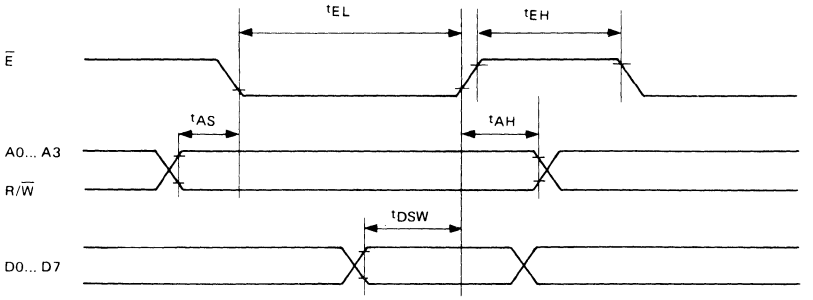
CLOCK AND OUTPUT CHARACTERISTICS



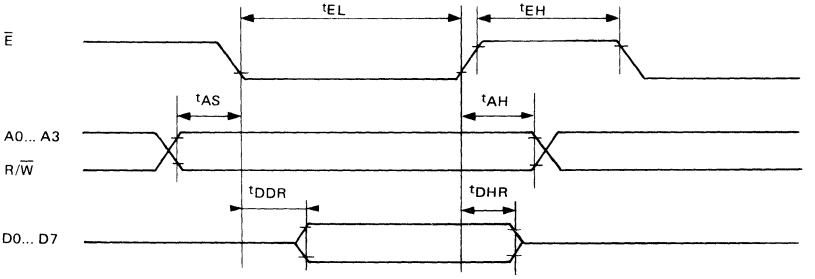
$\overline{IRQ}$ RELEASE TIME



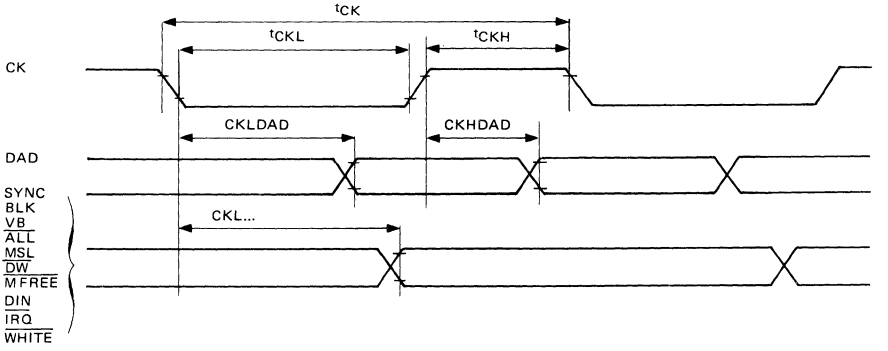
MICROPROCESSOR BUS, WRITE ACCESS



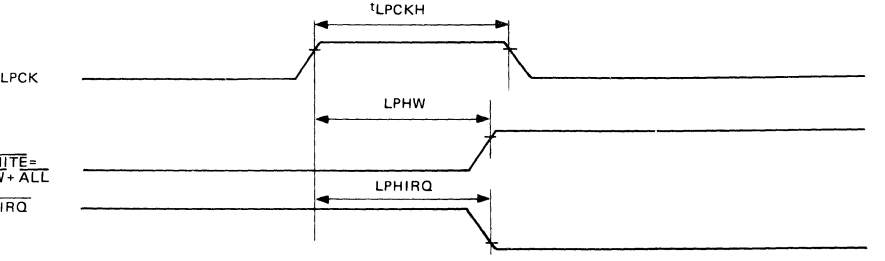
MICROPROCESSOR BUS, READ ACCESS



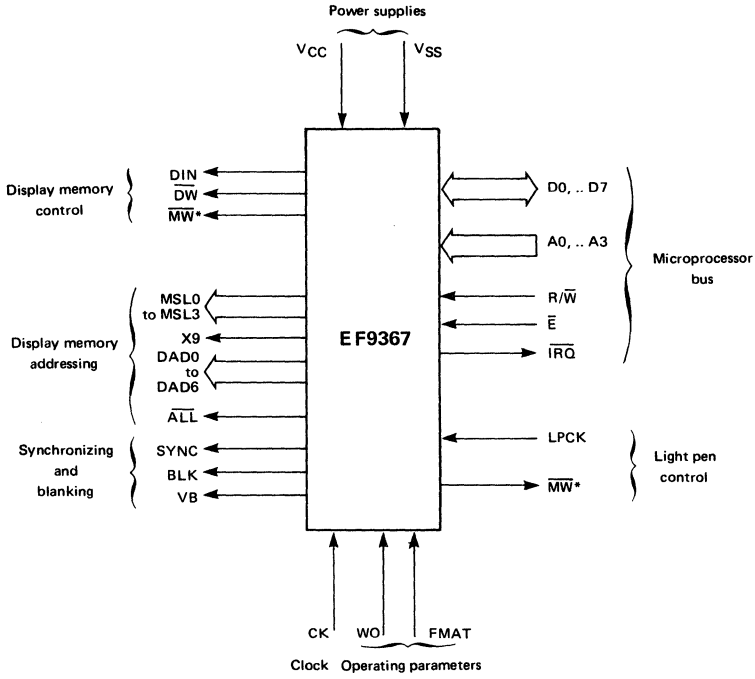
SYNCHRONOUS SIGNALS WITH CK INPUT



LIGHT PEN SIGNALS



PIN DESCRIPTION



*This pin outputs two items of data multiplexed by signal $\overline{ALL}$.

POWER SUPPLY, CLOCK AND OPERATING PARAMETERS

NAME	PIN TYPE	N°	FUNCTION	DESCRIPTION
V_{SS}	S	20	Power supply	Ground
V_{CC}	S	40	Power supply	+ 5 V
CK	I	1	Clock	Master clock. All internal processor states are modified on the falling edge of this signal. The whole circuit logic is static and the cycle of this clock needs only to be ajusted according to the shape and accuracy the synchronizing signals should feature. DAD memory address multiplexing signal. If CK is low, low addresses (or row addresses for the memory) are those that are output on DAD. The frequency of CK is a multiple of the image refresh frequency : - interlaced scanning : f (CK) = f (1/2 frame) x (625 or 525) x 96 - non-interlaced scanning : f (CK) = f (frame) x (312 or 262) x 96.
FMAT	I	8	Format	This pin is connected to V_{CC}, V_{SS}, CK or $\overline{CK}$ and sets the number of monitor and image lines : V_{CC} : 625 line monitor, interlaced synchronization, 512 lines displayed CK : 525 line monitor, interlaced synchronization, 416 lines displayed $\overline{CK}$: 525 line monitor, non-interlaced synchro, 208 lines displayed V_{SS} : 625 line monitor, non-interlaced synchro, 256 lines displayed.
WO	I	23	Write only	When WO is high, memory refresh nor display no longer exist. The hard wired write processors may operate without being interrupted. The $\overline{ALL}$ signal is always high.

SYNCHRONIZING AND BLANKING SIGNALS

SYNC	O	34	Video monitor synchronizing	Video monitor line and frame synchronization signal. For example, if CK is at 1.5 MHz and FMAT is high, signal SYNC is to CCIR 625 line 50 Hz standard. This output is independent of input WO and of register CTRL1.
BLK	O	25	Blanking	This signal is high apart from the display window (writing or refresh). It is always high if bit 2 in register CTRL1 is high, but it is not affected by the WO input.
VB	O	16	Vertical blanking	This signal is not affected by WO and register CTRL1. High during vertical blanking.

DISPLAY MEMORY ADDRESSING SIGNALS

DAD0 to DAD6	O	37,39, 38,4 3,2,5	Display address	Addresses that are multiplexed by the CK signal. Provided for the automatic refresh of the 16 K or 64 K dynamic memories.
X9	O	19	Memory address	Horizontal pointer extension bit for write operations (horizontal resolutions greater than 512).
MSL0 to MSL3	O	6,36 7,35	Memory select	Pixel write select signals (see section : **Display memory configuration.**)
$\overline{ALL}$	O	22	Access to all memory units	This signal makes it possible to discriminate between the collective memory accesses to all chips (display, refresh or erase), and the memory accesses to a single pixel for vector or character writing purposes. This signal is low for collective access.

DISPLAY MEMORY CONTROL SIGNALS

NAME	PIN TYPE	N°	FUNCTION	DESCRIPTION
DIN	O	15	Display in	Selection of the memory data code corresponding to the display screen in the 'off' condition (active when high). For a black-and-white display (1 bit per pixel), DIN may directly be the storage entry data.
$\overline{DW}$	O	14	Display write	Display memory write signal. Active when low.
$\overline{MW}$	O	24	Memory available	This pin outputs $\overline{MFREE}$ and $\overline{WHITE}$ signals which are externally demultiplexed by signal $\overline{ALL}$: $\overline{MFREE} = \overline{MW} + ALL$; $\overline{WHITE} = \overline{MW} + \overline{ALL}$ **Memory free ($\overline{MFREE}$) :** Signal low during the next memory idle period following the $0F_{16}$ command. This signal allows exchanges between the microprocessor and the X and Y flagged memory segment without affecting the display. **Forcing to white level ($\overline{WHITE}$) :** Forces white level on video signal, for use of the light pen. Active when low.

MICROPROCESSOR BUS SIGNALS

D0-D7	I/O	33 to 26	Data bus	I/O buffers opening is controlled through $\overline{E}$, and the related direction through R/$\overline{W}$.
A0-A3	I	9 to 12	Address bus	Address of the register involved in microprocessor access.
R/$\overline{W}$	I	18	Read/write signal	Read/write signal. Write when low.
$\overline{E}$	I	17	Enable	Bus exchange synchronizing and enabling signal.
$\overline{IRQ}$	O	13	Interrupt request	Interrupt request towards the microprocessor, programmable through register CTRL1. Open drain output.

LIGHT PEN OPERATING SIGNALS

LPCK	I	21	Light pen strobe	Light pen input. When the mechanism is set, a rising edge loads into registers XLP and YLP the current display address and sets the XLP register's LSB high.

REGISTER DESCRIPTION

X AND Y REGISTERS (Addresses : 8_{16}, 9_{16}, A_{16}, B_{16})

The X and Y registers are 12-bit read-write registers. They indicate the position of the next dot to be written into the display memory. They have no connection at all with the video signal generating scan, but they point the write address, in the same way as the pen address on a plotter.

These 2 registers are incremented or decremented, prior to each write operation into the display memory, by the internal vector and character generators, or they may be directly positioned by the microprocessor.

This 2 x 12 bit write address covers a 4096 x 4096 point addressing space. Only the LSBs are used here, since the maximum definition of the picture actually stored is 512 x 1024 pixels (picture elements).

In practice, the GDP assumes that it has a memory space of 1024 x 512 (FMAT = V_{CC} or CK) or 1024 x 256 (FMAT = V_{SS} or $\overline{CK}$) and disables writing outside this space, unless bit 3 of CTRL 1 is set.

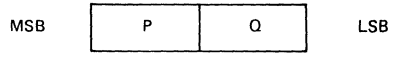
The above features along with the relative mode description of all picture component elements make it possible to automatically solve the great majority of edge cut-off problems.

DELTAX AND DELTAY REGISTERS (Addresses : 5_{16}, 7_{16}).

The DELTAX and DELTAY registers are 8-bit read-write registers. They indicate to the vector generator the projections of the next vector to be plotted, on the X and Y axes respectively. Such values are unsigned integers. The plotting of a vector is initiated by a write operation in the command register (CMD).

CSIZE REGISTER (Address : 3_{16})

The CSIZE register is an 8-bit read-write register. It indicates the scaling factors of X and Y registers for the symbols and characters. 98 characters are generated from a 5 x 8 pixel matrix defined by an internal ROM. In the standard version, it contains the alphanumeric characters in the ASCII code which may be printed, together with a number of special symbols.



Each symbol can be increased by a factor P(X) or Q(Y). These factors are independent integers which may each vary from 1 to 16 and which are defined by the CSIZE register. The symbol generation sequence is started after writing the ASCII code of the symbol to be represented in the CMD register.

CTRL1 REGISTER (Address : 1_{16}).

The CTRL1 register is a 7-bit read-write register, through which the general circuit operation may be fed with the required parameters.

Bit 0 : When low, this bit inhibits writing in display memory (equivalent to pen or eraser up).
When high, this bit enables writing in display memory (pen or eraser down).
This bit controls the $\overline{DW}$ output.

Bit 1 : When low, this bit selects the eraser.
When high, this bit selects the pen.
This bit controls the DIN output.

Bit 2 : When low, this bit selects normal writing mode (writing apart from the display and refresh periods, which are a requirement for the dynamic storages) in display memory.
When high, this bit selects the high speed writing mode : the display periods are deleted. Only the dynamic storage refresh periods are retained.

Bit 3 : When low, this bit indicates that the 4096 x 4096 space is being used (the 12 X and Y bits are significant)
When high, this bit selects the cyclic screen operating mode.

Bit 4 : When low, this bit inhibits the interrupt triggered by the light pen sequence completion.
When high, this bit enables the interrupt.

Bit 5 : When low, this bit inhibits the interrupt release by vertical blanking.
When high, this bit enables the interrupt.

Bit 6 : When low, this bit inhibits the interrupt indicating that the system is ready for a new command.
When high, this bit enables the interrupt.

Bit 7 : Not used. Always low in read mode.

CTRL2 REGISTER (Address : 2_{16})

The CTRL2 register is a 4-bit read/write register, through which the plotting of vectors and characters may be denoted by parameters.

Bit 0, 1 : These 2 bits define 4 types of lines (continuous, dotted, dashed, dash-dotted).

Bit 2 : When low, this bit defines straight writing.
When high, it defines tilted characters.

Bit 3 : When low, this bit defines writing along an horizontal line.
When high, this bit defines writing along a vertical line.

Bit 4, 5, 6, 7 : Not used. Always low in read mode.

CMD COMMAND REGISTER (Address : 0_{16})

The CMD register is an 8-bit write-only register. Each write operation in this register causes a command to be executed, upon completion of the time necessary for synchronizing the microprocessor access and the GDP's CK clock.

Several types of command are available :

- vector plotting
- character plotting
- screen erase
- light pen circuitry setting
- access to the display memory through an external circuitry.
- indirect modification of the other registers (commands that make it possible for the X, Y, DELTAX, DELTAY, CTRL1, CTRL2 and CSIZE registers to be amended or scratched).

STATUS REGISTER (Address 0_{16} or F_{16})

The STATUS register is an 8-bit read-only register. It is used to monitor the status of the executing statements entered into the circuit, and more specifically to avoid the need for modifying a register that is already used for the command currently executing.

Bit 0 : When low, this bit indicates that a light pen sequence is currently executing.
When high, it indicates that no light pen sequence is currently executing.

Bit 1 : This bit is high during vertical blanking. It is the VB signal recopy.

Bit 2 : When low, this bit indicates that a command is currently executing.
When high, this bit indicates that the circuit is ready for a new command.

Bit 3 : This bit when low indicates that registers X and Y are pointing within the assumed memory space.
This bit is obtained by applying the logical OR function to the unused most significant bits of registers X and Y.
If FMAT = V_{CC} or CK, the assumed memory space is 1024 x 512.
If FMAT = V_{SS} or $\overline{CK}$, the assumed memory space is 1024 x 256.

Bit 4 : When high, this bit indicates that an interrupt has been initiated by the completion of a light pen running sequence and that this interrupt has been enabled by bit 4 in CTRL1 register.

Bit 5 : When high, this bit indicates that an interrupt has been initiated by vertical blanking and that this interrupt has been enabled by bit 5 in CTRL1 register.

Bit 6 : When high, this bit indicates that an interrupt has been initiated by the completion of execution of a command and that this interrupt has been enabled by bit 6 in CTRL1 register.

Bit 7 : When high, this bit indicates that an interrupt has been initiated. It is the logic OR of bits 4, 5 and 6 in STATUS register. The $\overline{IRQ}$ output state is always the opposite of the status of this bit.

Note : Bits 4, 5, 6 and 7 are reset low by reading the STATUS register at address 0_{16}. Reading at address F_{16} does not modify their state.

XLP AND YLP REGISTERS (Addresses C_{16} and D_{16})

The XLP and YLP registers are read-only registers, with 7 and 8 bits respectively. Upon completion of a light pen running sequence, they contain the display address sampled by the first edge appearing rising on the LPCK input. The use of such registers is discussed in section : **Use of light pen circuitry.**

NOTES :

1. All internal registers may be read or written at any time by the microprocessor. However, the precautions outlined below should be observed :

   – Do not write into the CMD register if execution of the previous command is not completed (bit 2 of STATUS register).

   – Do not alter any register if it is used as an input parameter for the internal hardwired systems (e. g. : modifying the DELTAX register while a vector plotting sequence is in progress).

   – Do not read a register that is being asynchronously modified by the internal hardwired systems (e. g. : reading the X register while a vector plotting sequence is in progress may be erroneous if CK and $\overline{E}$ are asynchronous).

2. On powering up, the writing devices may have any status. Before entering a command for the first time, it is necessary to wait until all functions currently underway are completed, which information can be derived from the STATUS register.

SYSTEM OPERATING PRINCIPLE

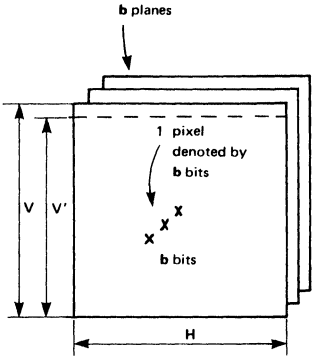
DISPLAY MEMORY CONFIGURATION

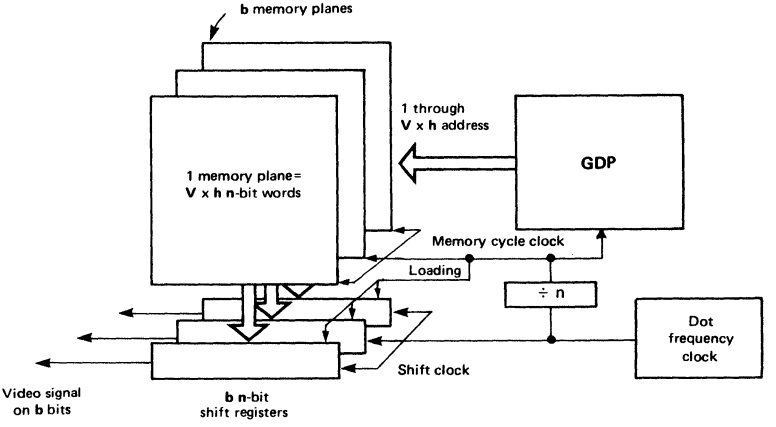
Assume a V x H pixel picture. Assume that each pixel is able to adopt 2^b different states. A V x H x b bit display memory is thus required.

In those applications where H features a high value, the video signal frequency exceeds the maximum frequency of memory read access.

Example : H = 512 with a television line frequency : the pixel succession period on the video signal is 83 ns.

It is mandatory that a line of H dots be cut into h adjoining segments of n bits each, read at the same time in the display memory, and thereafter converted to serial form to produce the video signal. h memory accesses per line are necessary. Each access loads b n-bit shift registers. The memory contains V x h x b n-bit words.





The EF9367 is designed for the following stored image formats :

- V = 512 or 256 (50 Hz)
- V' = 416 or 208 (60 Hz)
- H = h x n
- H = 1024 or lower multiples of 64
- h = 64
- n = 16, 8, 4, 2, 1 (or any value below 16 using external PROM encoding)
- b = any value (addressing is same for all memory planes, management of these planes is external to the GDP).

In so far as the overflow tests are concerned, the circuit assumes that it still has the maximum memory space for X (1024). The test for Y is effected in the following memory spaces :

512 if FMAT = V_{CC} or $\overline{CK}$
256 if FMAT = V_{SS} or $\overline{CK}$

512 or 256 vertical resolution : the displayed space is identical to the space in memory (unless a greater memory capacity is deliberately selected).

416 or 208 vertical resolution : the displayed space is smaller than the memory space.

Lines not displayed are displayable using an external adder to dejustify the display addresses (this arrangement may be used for smooth roll-up/roll down.

DAD AND MSL OUTPUT STATUS TABLES

The internal counters which address the display memory are made up of :

- 6 horizontal address bits (h = 64)
  $h_0, h_1, h_2, h_3, h_4, h_5$ (h_0 = LSB)
- **9 vertical address bits (V ⩽ 512)**
  $t, V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7$

t is here the LSB. It denotes the line parity and changes every frame because of interlaced scan. Within a same frame, V_0 denotes the LSB.

The write address is made up of the LSBs of the X and Y internal registers.

$$X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9$$
$$Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, Y_8$$

The GDP produces addressing signals in the sequences shown in the tables opposite :

FMAT = V_{CC} or CK

		MSL				X_9	DAD						
$\overline{ALL}$	CK	0	1	2	3		0	1	2	3	4	5	6
0	0	X_0	X_1	X_2	V_1	X_9	h_5	h_4	h_3	h_2	h_1	h_0	V_0
0	1						V_7	V_6	V_5	V_4	V_3	V_2	t
1	0	X_0	X_1	X_2	Y_2	X_9	X_8	X_7	X_6	X_5	X_4	X_3	Y_1
1	1						Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_0

FMAT = V_{SS} or $\overline{CK}$

		MSL				X_9	DAD						
$\overline{ALL}$	CK	0	1	2	3		0	1	2	3	4	5	6
0	0	X_0	X_1	X_2	1	X_9	h_5	h_4	h_3	h_2	h_1	h_0	V_0
0	1						V_7	V_6	V_5	V_4	V_3	V_2	V_1
1	0	X_0	X_1	X_2	1	X_9	X_8	X_7	X_6	X_5	X_4	X_3	Y_0
1	1						Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1

DESCRIPTION OF DISPLAYABLE FORMATS

NON INTERLACED SCANNING

256 x 512 or 208 x 512 pixel formats (H = 512, n = 8)

Input FMAT must be low or connected to $\overline{CK}$.
The memory is made up of 16 K bytes per memory plane. The byte address is made up of 14 bits which are output on two runs on the DAD pins. The three MSL0, MSL1, MSL2 outputs are used to select one pixel out of the eight featuring the same address. They issue the number of the pixel, encoded on three bits. MSL3 is high, and is not used.

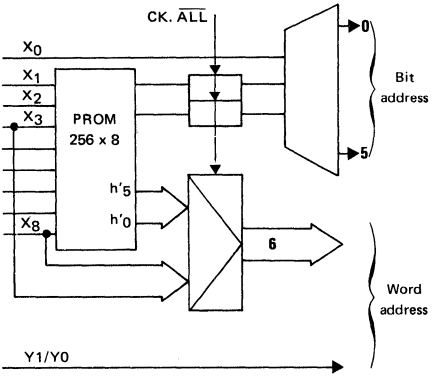
256 x 384 or 208 x 384 pixel formats (H = 384, n = 6)

Input FMAT must be low or connected to $\overline{CK}$.
The memory is organized as 16 K words x 6 bits.

The signals produced by the chip in the sequence indicated for the 256 x 512 format are transcoded externally as shown in the opposite diagram.

256 x 320 or 208 x 320 pixel formats (H = 320, n = 5)

The same schematic as for 384 horizontal resolution should be used with a memory organized in 5 bit words.



256 x 256 or 208 x 256 pixel formats (H = 256, n = 4)

Input FMAT must be low or connected to $\overline{CK}$.
The memory is made up of 16 K words x 4 bits. The word address is made up of 14 bits which are output in two runs on the DAD pins. One of the four chips is selected by decoding pins MSL1 and MSL2 (that leads to ignore X_0 : the X computation space is changed to 2048 pixels and horizontal overflow detected at 512 pixels).

INTERLACED SCANNING

512 x 1024 or 416 x 1024 pixel formats (H = 1024, n = 16)

Input FMAT must be connected to V_{CC} or CK.
The memory comprises 32 K words x 16 bits, organized in two blocks of 16 K words each.

The signals produced by the circuit in the sequence indicated for the 512 x 512 format are combined externally as shown at the end of the data sheet.

512 x 768 or 416 x 768 pixel formats (H = 768, n = 12)

Input FMAT must be connected to V_{CC} or CK.
The memory comprises 32 K words x 12 bits, organized in two blocks of 16 K words each.

The signals produced by the chip in the sequence indicated for the 512 x 512 format are transcoded externally as shown in the diagram below.

512 x 640 or 416 x 640 pixel formats (H = 640, n = 10)

The same schematic as below should be used with a memory organized in 10 bit words.

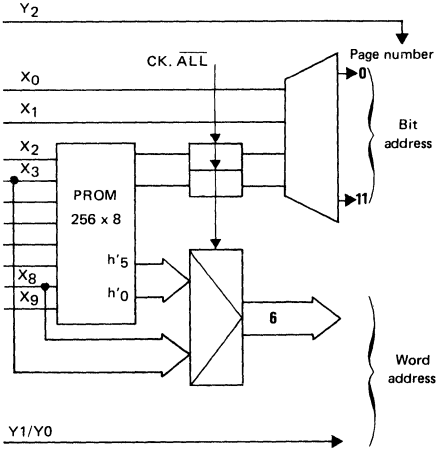
512 x 512 or 416 x 512 pixel formats (H = 512, n = 8)

The FMAT input should be tied to V_{CC} or CK. The memory is made up of V x h bytes = 32 K bytes per memory plane.

The byte address is made up of 15 bits :

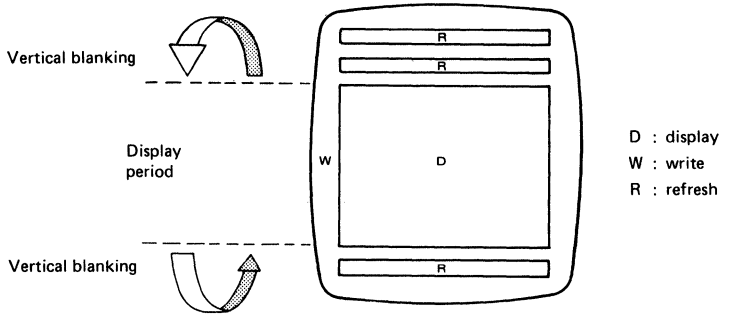
- 14 are output in 2 runs on the DAD pins for the purpose of using 16 K x 1 bit dynamic RAMs,
- the 15th one is output on pin MSL3.

The 3 MSL0, 1 and 2 outputs allow to select one pixel out of the 8 featuring the same address, for pixel-to-pixel write applications. They issue the number of the involved pixel, encoded on 3 bits.



MEMORY OPERATION SEQUENCE ALONG ONE FRAME

Apart from the window where the memory is used for display purposes exclusively, write operations may be performed, except during 3 refresh periods.



The three period types, D, W and R, respectively, are indicated outside the circuit through the BLK and $\overline{ALL}$ signals :

	BLK	$\overline{ALL}$
D	0	0
W	1	1
R	1	0

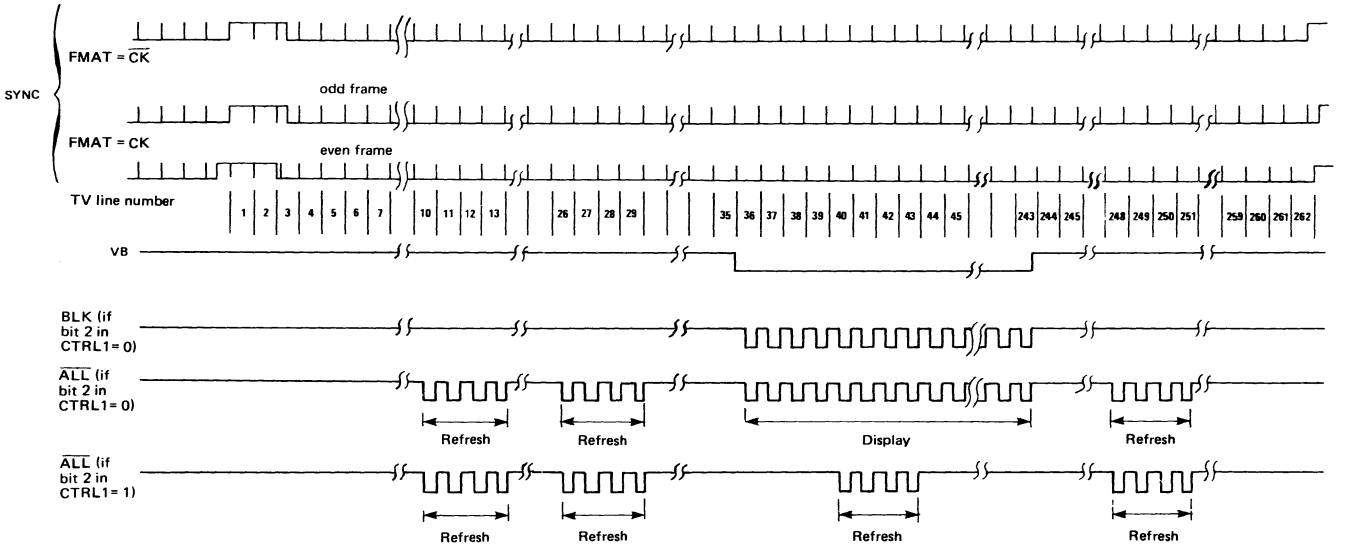
The refresh of dynamic RAMs is automatically performed by the GDP. During display, the memory is entirely refreshed each 4 lines (256 accesses).
During vertical blanking, 3 refresh cycles of 4 lines each are executed.

Exceptions :

- If bit 2 in register CTRL1 is high (high speed write), the display period is suppressed and 19 refresh cycles of 4 lines each are executed during one frame.
- As long as the WO input is high, the circuit is set to write mode, and BLK retains the same outline as it has under normal operating conditions.

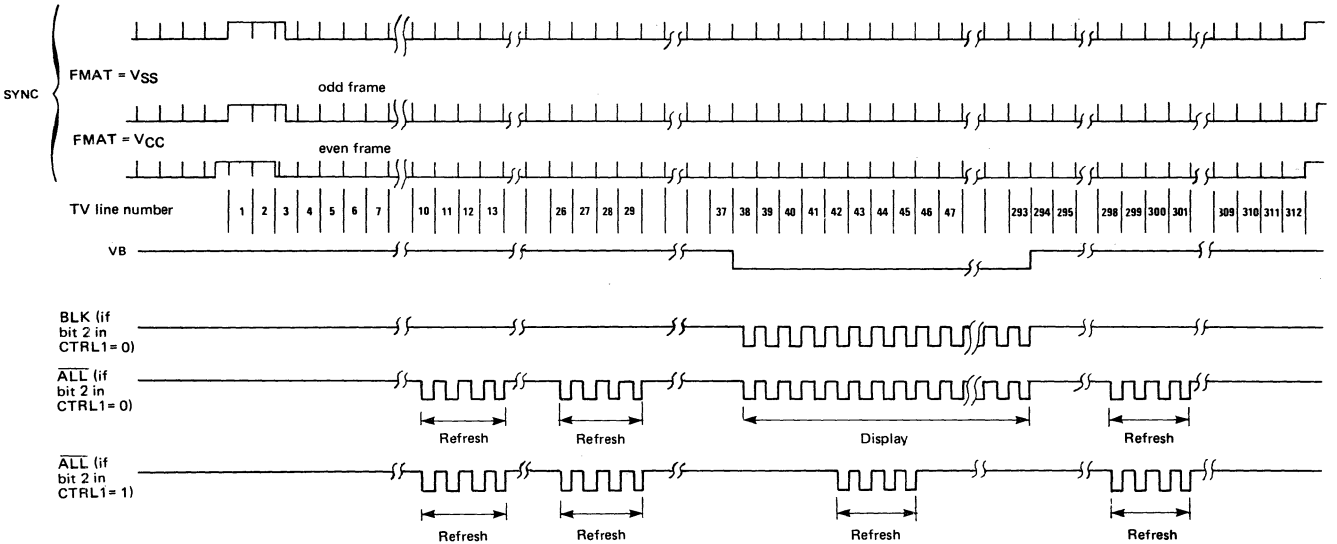
  In these two cases, executing codes 04_{16}, 06_{16}, 07_{16} and $0C_{16}$ triggers a complete D sequence for a high-speed scan of all addresses. This lasts two frames if FMAT is high (or tied to CK) and one frame if FMAT is low (or tied to $\overline{CK}$).

FRAME SEQUENCE – 525 LINE SYNCHRONIZATION



Note : $\overline{ALL}$ signal high denotes write periods.

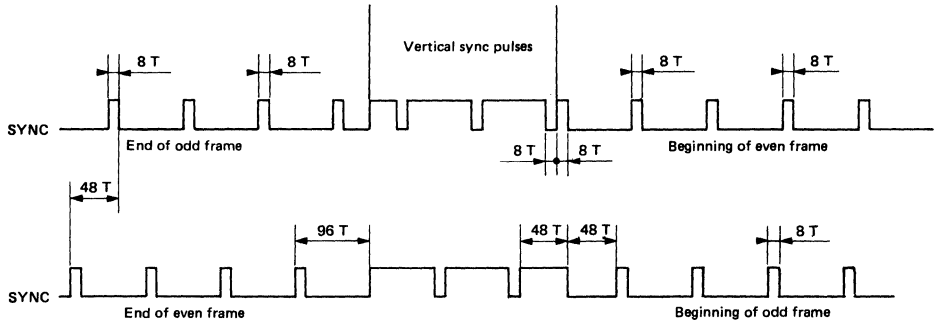
FRAME SEQUENCE – 625 LINE SYNCHRONIZATION



Note : $\overline{ALL}$ signal high denotes write periods.

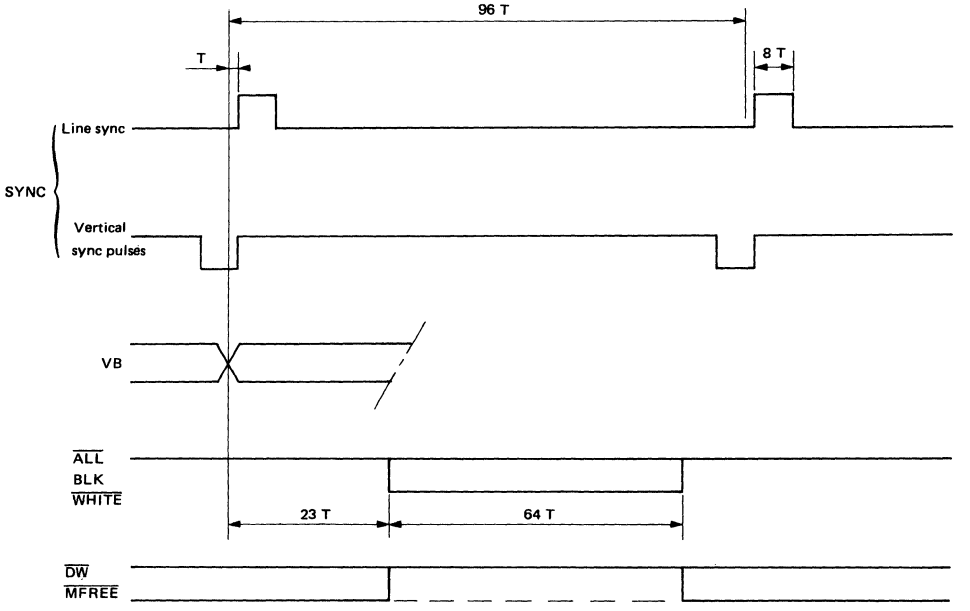
COMPOSITE SYNC AROUND FRAME SYNC

T : CK input period (667 ns in typical application where TV line duration is 64 μs)



Note : If FMAT is low or tied to $\overline{CK}$, the pattern of the second line is repeated for each frame.

DETAILED LINE DIAGRAM



HARDWIRED WRITE PROCESSOR OPERATION IN DISPLAY MEMORY

The hardwired write processors are sequenced by the master clock CK. They receive their parameters from the microprocessor bus. They control the X, Y write address, and the DIN, $\overline{DW}$, $\overline{MW}$ and $\overline{IRQ}$ outputs.

These harwired processors operate in continuous mode. In the event of conflicting access to the display memory, the display and refresh processors have priority.

Since command decoding is synchronous with the CK master clock, any write operation into the (CMD) command register triggers a synchronizing mechanism which engages the circuit for a maximum of 2 CK cycles when the $\overline{E}$ input returns high. The circuit remains engaged throughout command execution.

No further command should be entered as long as bit 2 in STATUS register is low.

VECTOR PLOTTING

The internal vector generator makes it possible to modify, within the display memory, all the dots which form the approximation of a straight line segment. All vectors plotted are described by the origin dot and the projections on the axes.

The starting point co-ordinates are defined by the X, Y register value, prior to the plotting operation.
Projections onto the axes are defined as absolute values by the DELTAX and DELTAY registers, with the sign in the command byte that initiates the vector plotting process.

The vector approximation achieved here is that established by J. F. BRESENHAM ("Algorithm for computer control of a digital plotter"). This algorithm is executed by a hardwired processor which allows for a further vector component dot to be written in each CK clock cycle.

During plotting, the display memory is addressed by the X, Y registers, which are incremented or decremented.

On completion of vector plotting, they point to the end of this vector.

All vectors may be plotted using any of the following line patterns : continuous, dotted, dashed, dash-dotted, according to the 2 LSBs in register CTRL2.

Irrespective of such patterns, the plotting speed remains unchanged. The "pen down-pen up" statement required for plotting non-continuous lines is controlled by the $\overline{DW}$ output.

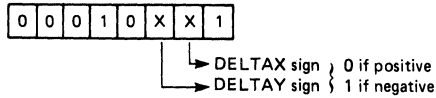
For a specified non-continuous line plotted vector, defined by DELTAX, DELTAY, CTRL2, CMD, the $\overline{DW}$ sequencing during the plotting process is always the same, irrespective of vector origin and of the nature of previous plots. This feature guarantees that a specified vector can be deleted by plotting it again after moving X and Y to the starting point, and complementing bit 1 in register CTRL1.

Since the vector plotting initiation command defines the sign of the projections onto the axes, all vectors may be plotted using 4 different commands.

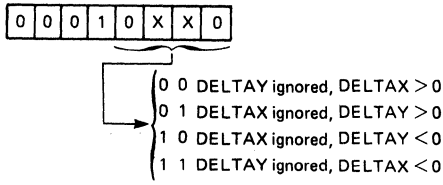
For increased programming flexibility, the system incorporates 16 different commands, supplemented by a set of 128 commands which make it possible to plot small size vectors by ignoring the DELTAX and DELTAY registers.

Such commands are as follows :

- Basic commands

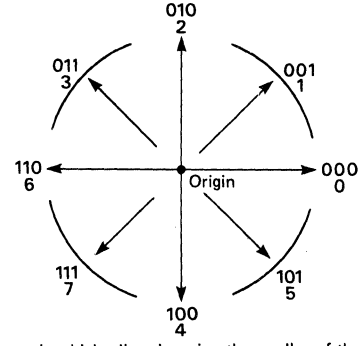


- Commands which allow ignoring the DELTAX or DELTAY registers by considering them as of zero value.

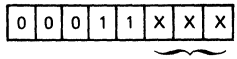


Note : Bits 1 and 2 always have the same sign meaning.

These 8 codes may be summarized by the following diagram :

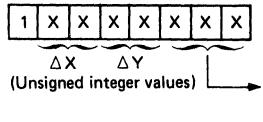


- Commands which allow ignoring the smaller of the two DELTAX and DELTAY registers, by considering it as being equal to the larger one, which is the same as plotting vectors parallel to the axes or diagonals, using a single DELTA register.



Same direction codes as above.

- Commands in which the two registers DELTAX and DELTAY may be ignored by specifying the projections through the CMD register (0 to 3 steps for each projection).



EXAMPLE : PLOTTING A DOTTED VECTOR

Origin : $\begin{cases} X = 47_{10} \\ Y = 75_{10} \end{cases}$

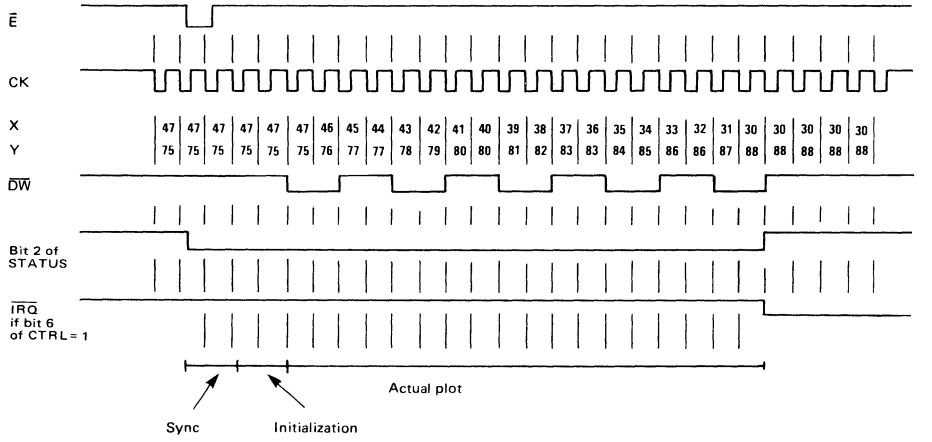
Projections: $\begin{cases} DELTAX = 17_{10} \\ DELTAY = 13_{10} \end{cases}$

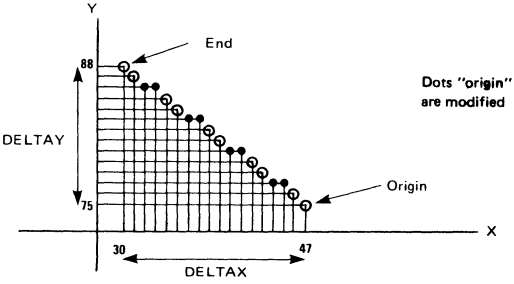
CMD = 13_{16} Corresponding to
- Basic command,
- DELTAX < 0
- DELTAY > 0

CTRL1 = 03_{16} Pen down

CTRL2 = 1_{16} Dotted vector :
2 dots on,
2 dots off.

Plotting cycle sequence : (It is assumed that the vector generator is not interrupted by the display or refresh cycle).





Dots "origin" and "end" are modified

Display shows : ● unmodified dot
o dot on

Note :

Plotting a vector with DELTAX = DELTAY = 0 writes the dot X, Y in memory. It occupies the vector generator for synchronization, initialization and one write cycle.

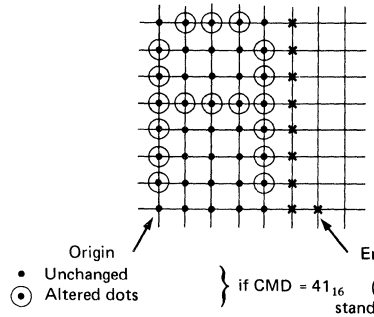
CHARACTER AND SYMBOL GENERATOR

The character generator operates in the same way as the vector generator, i.e. through incrementing or decrementing the X, Y registers, in conjunction with a $\overline{DW}$ output control.

It receives parameters from the CSIZE, CTRL2 and CMD registers. The characters plotted are selected, according to the CMD value, out of 98 matrices (97 8-dot high x 5-dot wide rectangular matrices, and one 4 dot x 4 dot matrix) defined in an internal ROM. Two scaling factors may be applied to the characters plotted using X and Y defined by the CSIZE register. The characters may be tilted, according to the content of register CTRL2.

Basic matrix

Upon completion of a character writing process, the X and Y registers are positioned for writing a further character next to the previous one, with a 1 dot spacing, i.e. Y is restored to its original value and X is incremented by 6.



× Computed dots, not defined into the ROM (not modifiable).

Scaling factors

Each individual dot in the 5 x 8 basic matrix may be replaced by a P x Q size block.

P : X co-ordinate scaling factor

Q : Y co-ordinate scaling factor

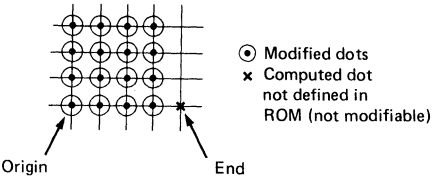
The character size becomes 5P x 8Q. Upon completion of the writing process, X is incremented by 6P. The CK clock cycle count required is 6P x 8Q.

P and Q may each take values from 1 through 16. They are defined by the CSIZE register. Each value is encoded on 4 bits, value 16 being encoded as 0_{16}.

In register CSIZE, P is encoded on the 4 MSBs and Q on the 4 LSBs.

Among the 97 rectangular matrices available in the standard ROM, 96 correspond to CMD values ranging from 20_{16} to $7F_{16}$, and the 97th matrix to $0A_{16}$. In the standard version, these values correspond to the 96 printable characters in the ASCII set. The 97th character is a 5P x 8Q block which may be used for deleting the other characters.

The 98th code ($0B_{16}$) is used to plot a 4P x 4Q graphic block. It locates X, Y, without spacing for the next symbol. Such a block makes it possible to pad uniform areas on the screen.



Tilted characters

All characters may be modified to produce tilted characters or to mark the vertical co-ordinate with straight or tilted type symbols. Such changes may be achieved using bits 2 and 3 in register CTRL2.

Note : Scaling factors P and Q are always applied within the co-ordinates of the character before conversion.

Character deletion

A character may be deleted using either the same command code or command code $0A_{16}$. In either case, bit 1 in register CTRL1 should be inverted, the origin should be the same as prior to a character plotting operation, as should the scaling factors.

Note : Vector generator and character generator operate in similar ways :

	VECTOR	CHARACTER
Dimensions	DELTAX, DELTAY	CSIZE, tilting
$\overline{DW}$ modulation	Type of line	Character code

USE OF LIGHT PEN CIRCUITRY

A rising edge on the LPCK input is used to sample the current display address in the XLP and YLP registers, provided that this edge is present in the frame immediately following loading of the 08_{16} or 09_{16} code into the CMD register.

Here, the frame origin is counted starting with the VB falling edge. With code 08_{16}, the $\overline{MW}$ output recopies the BLK signal from the frame origin up to the rising edge on the LPCK input, or when VB starts rising again, if the LPCK input remains low for the entire frame. With code 09_{16}, the $\overline{MW}$ output is not activated.

The YLP address is 8-bit coded since there are 256 display lines in each frame. The XLP address is 6-bit coded since there are 64 display cycles in each line.

These 6 bits left-justified in register XLP indicate the number of the segment (h = 0 to 63) to which the point indicated by the light pen belongs.

The address sampled into XLP corresponds to the current memory cycle. Dots detected by the light pen were addressed in the memory during the previous cycle. Hence, 1 should be subtracted from bit 2 in XLP register where the light pen electronic circuitry does not produce any additional delay.

If the rising edge on input LPCK occurs while VB is low, then the LSB in XLP is set high. This bit acts as a status signal which is reset to the low state by reading register XLP or YLP.

The rising edge first received (LPCK or VB) sets bit 0 in STATUS register high. An interrupt is initiated if bit 4 in CTRL1 is high.

When commands 08_{16} or 09_{16} have been decoded, bit 2 of the status register goes high (circuit ready for any further command) and bit 0 goes low (light pen operating sequence underway).

SCREEN BLANKING COMMANDS

Three commands (04_{16}, 06_{16}, 07_{16}) will set the whole display memory to a status corresponding to a "black display screen" condition. Another command ($0C_{16}$) may be used to set the whole memory to a status other than black (this condition being determined by bit 1 in register CTRL1).

The 4 commands outlined above use the planned scanning of the memory addresses achieved by the display stage. The X and Y registers are not affected by commands 04_{16} and $0C_{16}$. Hence, the time required is that corresponding to one frame (FMAT = 0 or $\overline{CK}$) or two frames (FMAT = 1 or CK). The time corresponding to the completion of the frame currently executing when the CMD register is loaded, should be added to the above time.

For the screen blanking process, the frame origin is counted starting with the VB falling edge.
The only signals affected here are the $\overline{DW}$ output, which remains low when VB is low, and the DIN output which is forced high where the 04_{16}, 06_{16} and 07_{16} commands are entered.

Such commands are activated without requiring action by WO input or bit 2 in register CTRL1. While these commands are executing, bit 2 in STATUS register remains low.

EXTERNAL REQUEST FOR DISPLAY MEMORY ACCESS ($\overline{MW}$ OUTPUT)

On writing code $0F_{16}$ into the CMD register, the $\overline{MW}$ output is set low by the circuitry, during the next free memory cycle.

Apart from the display and refresh periods, this cycle is the first complete cycle that occurs after input $\overline{E}$ is reset high.

During this cycle, those addresses output on DAD and MSL correspond to the X and Y register contents : $\overline{DW}$ is high, $\overline{ALL}$ is high.

Should the memory be engaged in a display or refresh operation, (which is the case when $\overline{ALL}$ is low), then this cycle is postponed to be executed after $\overline{ALL}$ is reset high. The maximum waiting time is thus 64 cycles.

The $\overline{MW}$ signal may be used e. g. for performing a read or write operation into a register located between the display memory and the microprocessor bus.

INTERRUPTS OPERATION

An interrupt may be initiated by three situations denoted by internal signals :

- Circuit ready for a further command
- Vertical blanking signal
- Light pen sequence completed.

These three signals appear in real time in the STATUS register (bits 0, 1, 2). Each signal is cross-referenced to a mask bit in the register CTRL1 (bits 4, 5, 6).

If the mask bit is high, the first rising edge that occurs on the interrupt initiating signal sets the related interrupt flip-flop circuit high.

The outputs from these three flip-flop circuits appear in the STATUS register (bits 4, 5, 6). If one flip-flop circuit is high, bit 7 in the STATUS register is high, and pin $\overline{IRQ}$ is forced low.

A read operation in the STATUS register at address 0_{16} resets its 4 MSBs low, after input $\overline{E}$ is reset high (a read at address F_{16} maintains their value).

The three interrupt control flip-flops are duplicated to prevent the loss of an interrupt coming during a read cycle of the STATUS register.

The status of bits 4, 5 and 6 corresponds to the interrupt control flip-flop circuit output, before input $\overline{E}$ goes low.

An interrupt coming during a read cycle of the STATUS register does not appear in bits 4, 5 and 6 during this read sequence, but during the following one. However, it may appear in bits 0, 1, 2 or on pin $\overline{IRQ}$.

TABLE 1 – REGISTER ADDRESS

ADDRESS REGISTER					REGISTER FUNCTIONS		Number of bits
Binary				Hexa	Read R/$\overline{W}$ = 1	Write R/$\overline{W}$ = 0	
A3	A2	A1	A0				
0	0	0	0	0	STATUS	CMD	8
0	0	0	1	1	CTRL 1 (Write control and interrupt control)		7
0	0	1	0	2	CTRL 2 (Vector and symbol type control)		4
0	0	1	1	3	CSIZE (Character size)		8
0	1	0	0	4	Reserved		—
0	1	0	1	5	DELTAX		8
0	1	1	0	6	Reserved		—
0	1	1	1	7	DELTAY		8
1	0	0	0	8	X MSBs		4
1	0	0	1	9	X LSBs		8
1	0	1	0	A	Y MSBs		4
1	0	1	1	B	Y LSBs		8
1	1	0	0	C	XLP (Light-pen)	Reserved	7
1	1	0	1	D	YLP (Light-pen)	Reserved	8
1	1	1	0	E	Reserved		—
1	1	1	1	F	STATUS	Reserved	8

Reserved : These addresses are reserved for future versions of the circuit. In read mode, output buffers D0-D7 force a high state on the data bus.

TABLE 2 – COMMAND REGISTER

b7 b6 b5 b4 / b3 b2 b1 b0			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0 0	0	Set bit 1 of CTRL1 : Pen selection	Vector generation (for b2, b1, b0 see small vector definition)		SPACE	0	@	P	`	p								
0 0 0 1	1	Clear bit 1 of CTRL 1 : Eraser selection			!	1	A	Q	a	q								
0 0 1 0	2	Set bit 0 of CTRL1 : Pen/Eraser down selection			"	2	B	R	b	r								
0 0 1 1	3	Clear bit 0 of CTRL 1 : Pen/Eraser up selection			#	3	C	S	c	s								
0 1 0 0	4	Clear screen			$	4	D	T	d	t								
0 1 0 1	5	X and Y registers reset to 0			%	5	E	U	e	u								
0 1 1 0	6	X and Y reset to 0 and clear screen			&	6	F	V	f	v								
0 1 1 1	7	Clear screen, set CSIZE to code "minsize" All other registers reset to 0 (except XLP, YLP)			'	7	G	W	g	w								
1 0 0 0	8	Light-pen initialization ($\overline{WHITE}$ forced low)	Special direction vectors (for b2, b1, b0 see small vector definition)		(	8	H	X	h	x								
1 0 0 1	9	Light-pen initialization			)	9	I	Y	i	y								
1 0 1 0	A	5 x 8 block drawing (size according to CSIZE)			*	:	J	Z	j	z								
1 0 1 1	B	4 x 4 block drawing (size according to CSIZE)			+	;	K	[	k	{								
1 1 0 0	C	Screen scanning : Pen or Eraser as defined by CTRL1			,	<	L	\	l	¦								
1 1 0 1	D	X register reset to 0			–	=	M	]	m	}								
1 1 1 0	E	Y register reset to 0			.	>	N	↑	n	¬								
1 1 1 1	F	Direct image memory access request for the next free cycle.			/	?	O	←	o	▒								

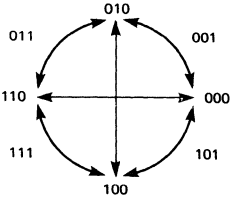
SMALL VECTOR DEFINITION :

b7	b6 b5	b4 b3	b2 b1 b0
1	|ΔX|	|ΔY|	Direction

Dimension

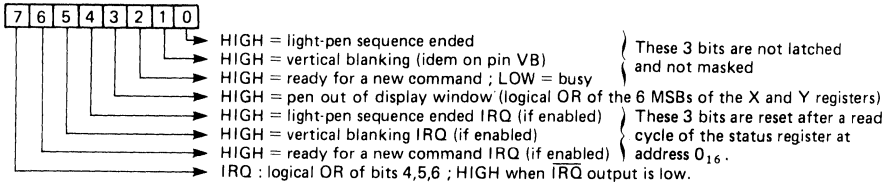
ΔX or ΔY	Vector length
0 0	0 step
0 1	1 step
1 0	2 steps
1 1	3 steps

Direction

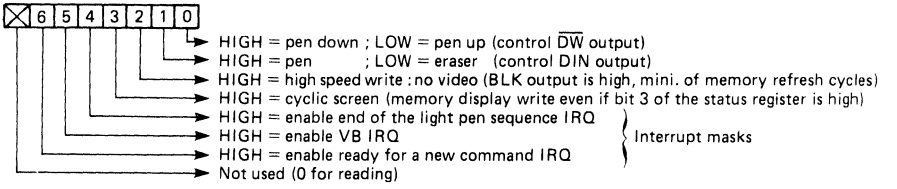


OTHER REGISTERS

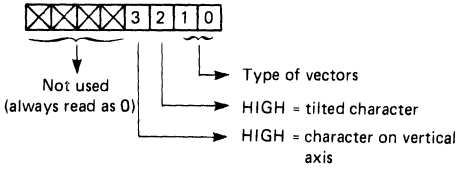
STATUS REGISTER (Read only)



CONTROL REGISTER 1 (Read/Write)

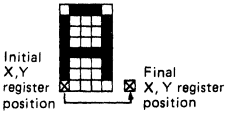


CONTROL REGISTER 2 (Read/Write)

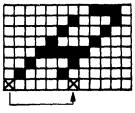


b1	b0	Type of vectors	
0	0	——— continuous	
0	1	- - - - - - dotted	2 dots on, 2 dots off
1	0	— — — dashed	4 dots on, 4 dots off
1	1	—·—·— dotted-dashed	10 dots on, 2 dots off, 2 dots on, 2 dots off.

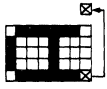
Types of character orientations



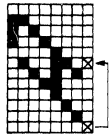
$b_3 = 0, b_2 = 0$
CSIZE = 11_{16}



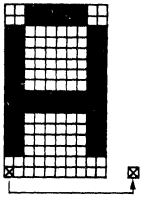
$b_3 = 0, b_2 = 1$
CSIZE = 11_{16}



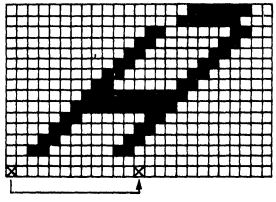
$b_3 = 1, b_2 = 0$
CSIZE = 11_{16}



$b_3 = 1, b_2 = 1$
CSIZE = 11_{16}

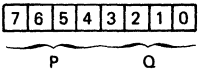


$b_3 = 0, b_2 = 0$
CSIZE = 22_{16}



$b_3 = 0, b_2 = 1$
CSIZE = 22_{16}

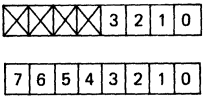
C-SIZE REGISTER (Read/Write)



P : Scaling factor on X axis
Q : Scaling factor on Y axis

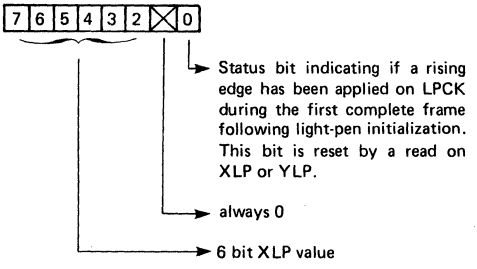
P and Q may take any value between 1 and 16. This value is given by the leftmost or rightmost 4 bits for P and Q respectively. Binary value (0) means 16.

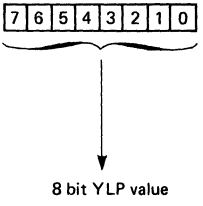
X AND Y REGISTERS (Read/Write)



The 4 leftmost MSBs are always 0.

XLP and YLP REGISTERS

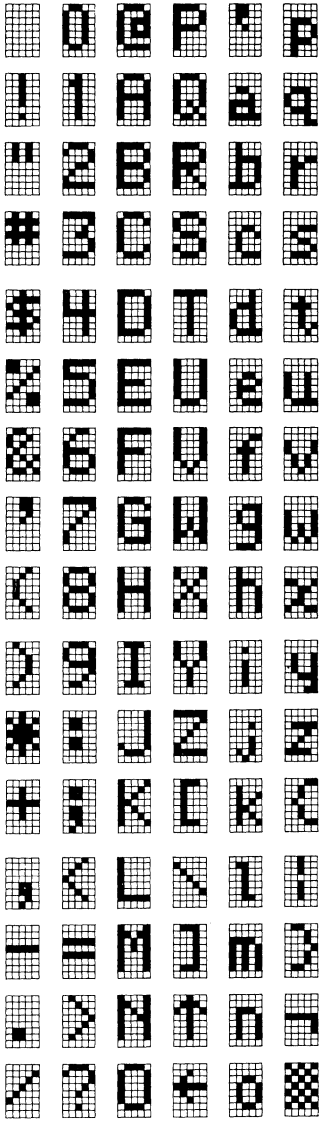




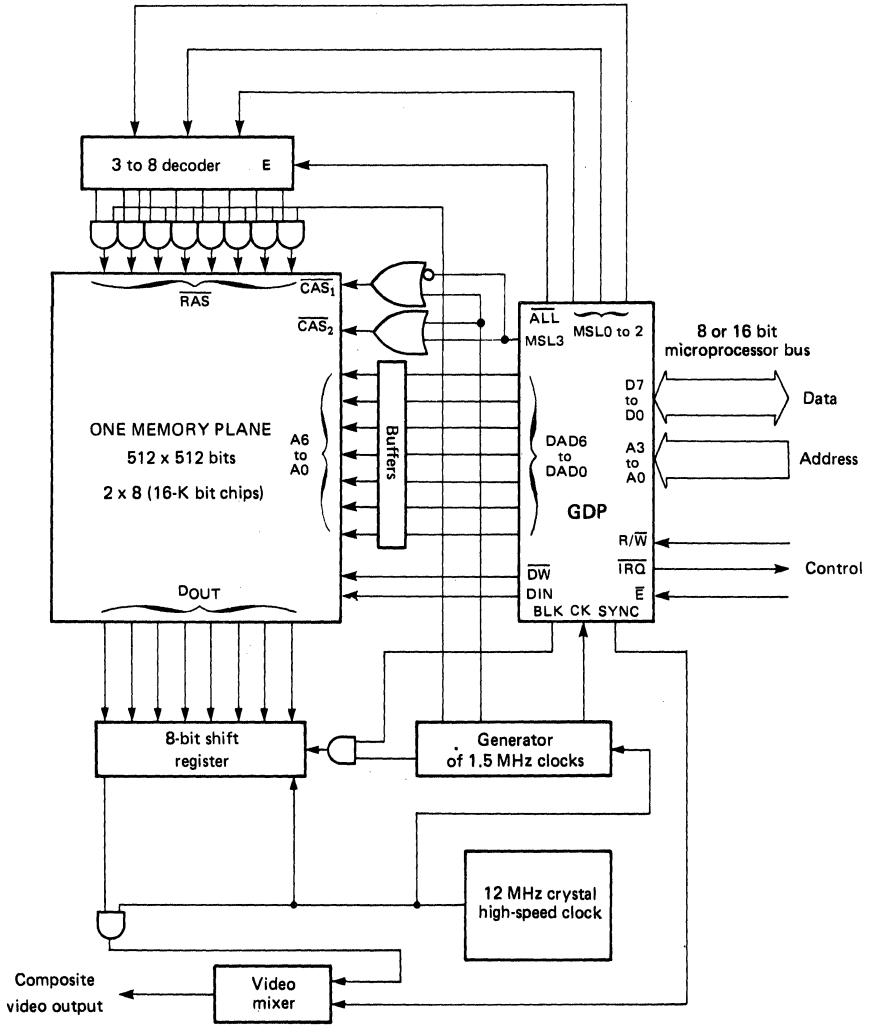
ASCII CHARACTER GENERATOR (5 x 8 matrix)

b7	0	0	0	0	0	0
b6	0	0	1	1	1	1
b5	1	1	0	0	1	1
b4	0	1	0	1	0	1

b3	b2	b1	b0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1



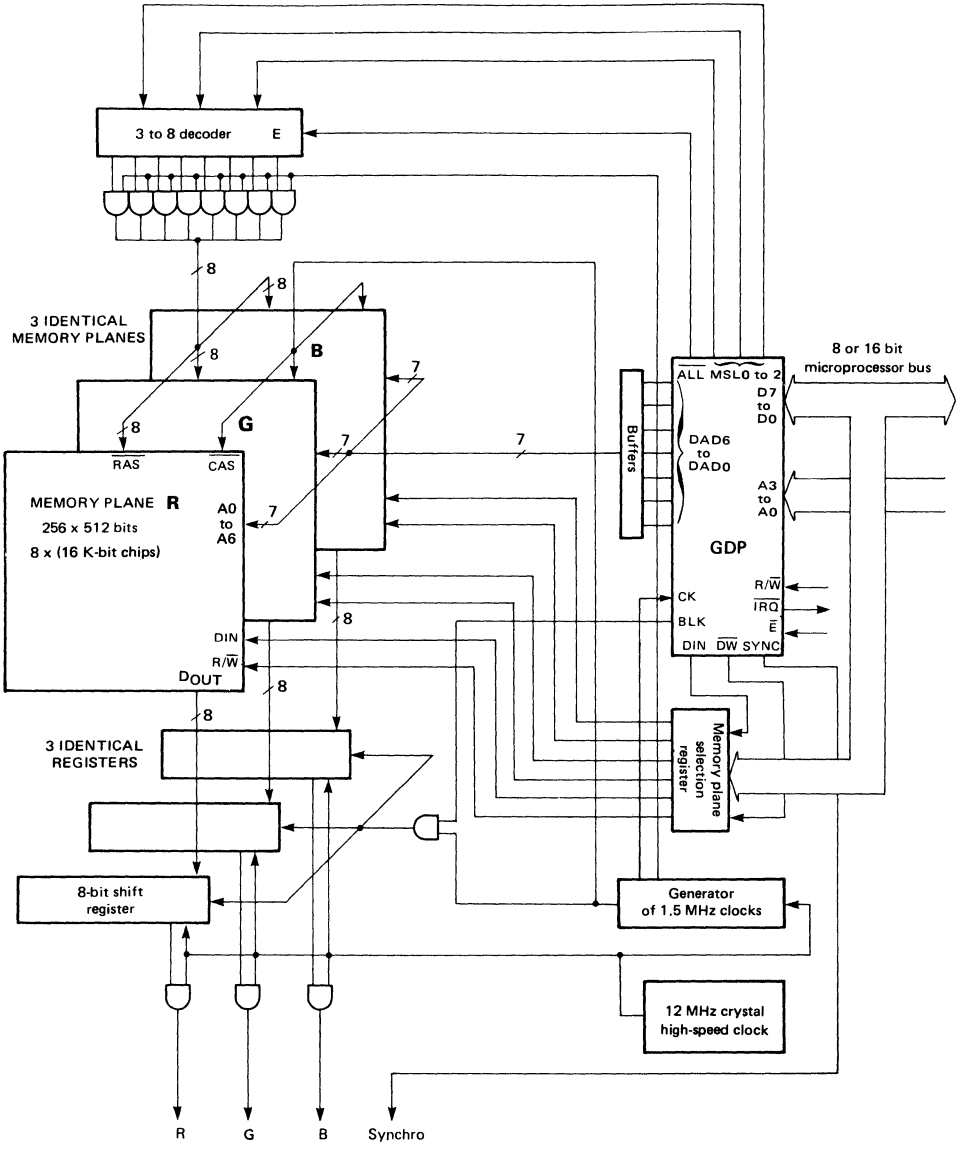
EXAMPLE OF A MONOCHROME APPLICATION : 512 x 512 or 416 x 512



Note : FMAT = V_{CC} : 512 x 512 resolution - 50 Hz 625 line interlaced scanning
FMAT = CK : 416 x 512 resolution - 60 Hz 525 line interlaced scanning.

EXAMPLE OF A COLOR APPLICATION : 208 x 512 or 256 x 512

Eight colours may be obtained from the three basic colours red (R), green (G), blue (B)

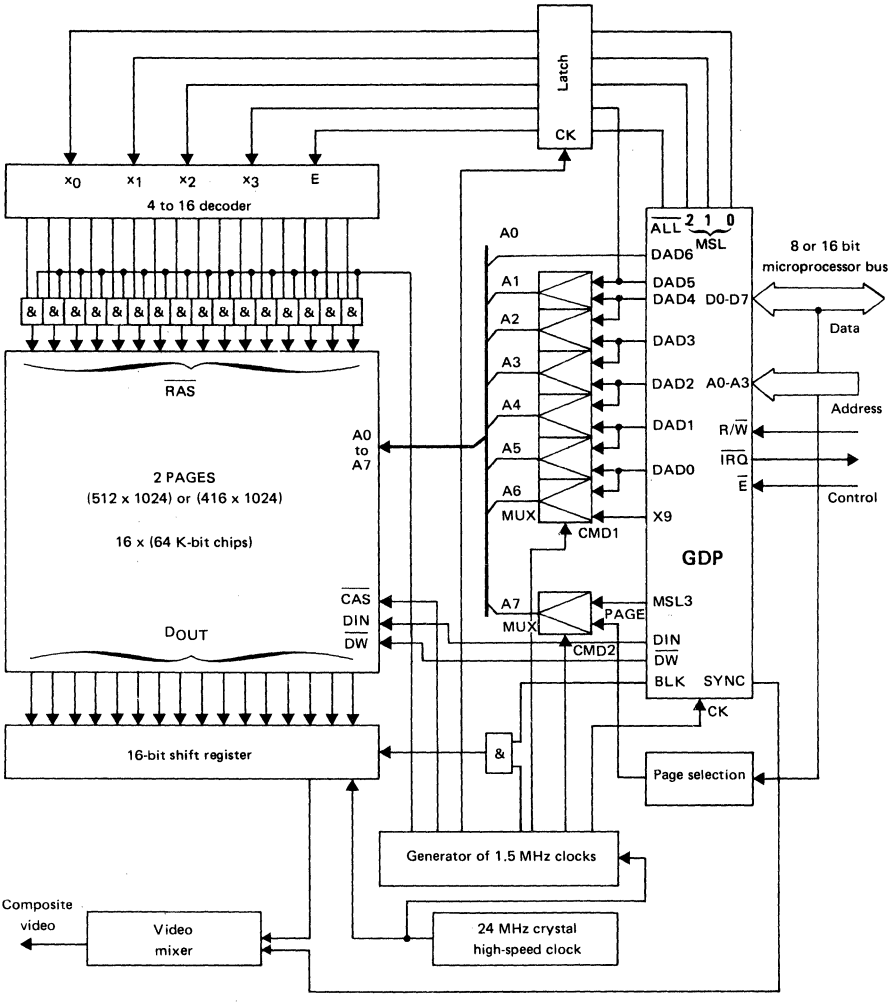


Note : FMAT = V_{SS} : 256 x 512 resolution - 50 Hz 625 line non interlaced scanning

FMAT = $\overline{CK}$: 208 x 512 resolution - 60 Hz 525 line non interlaced scanning.

EXAMPLE OF A MONOCHROME, MULTIPAGE APPLICATION : 512 x 1024 or 416 x 1024

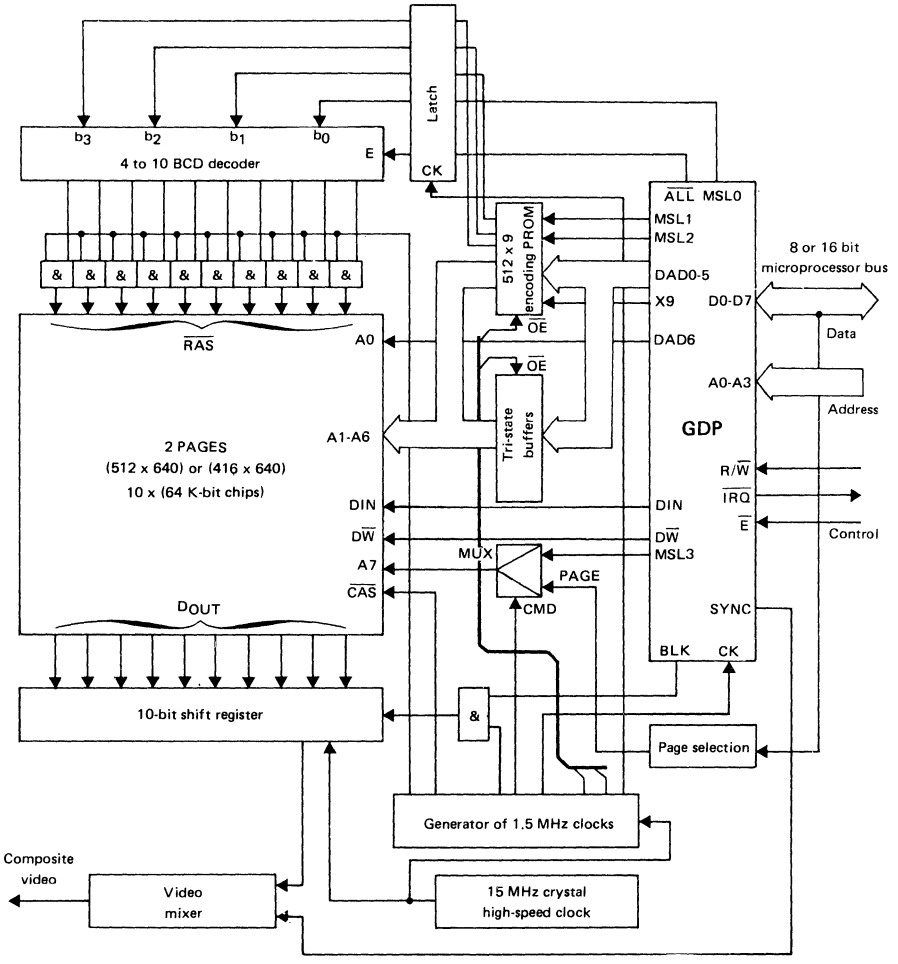
(See page 30 for MUX command law)



Note : FMAT = V_{CC} : 512 x 1024 resolution - 50 Hz 625 line interlaced scanning
FMAT = CK : 416 x 1024 resolution - 60 Hz 525 line interlaced scanning.

EXAMPLE OF A MONOCHROME, MULTIPAGE APPLICATION : 512 x 640 or 416 x 640

(See page 30 for PROM encoding)



Note : FMAT = V_{CC} : 512 x 640 resolution - 50 Hz 625 line interlaced scanning.
FMAT = CK : 416 x 640 resolution - 60 Hz 525 line interlaced scanning.

MUX COMMAND LAW

Following table indicates MUX command principles.

Selected MUX input				Output	Comment
Read cycles		Write cycles		Address bit	
$\overline{RAS}$	$\overline{CAS}$	$\overline{RAS}$	$\overline{CAS}$		
DAD6	DAD6	DAD6	DAD6	A_0	No MUX
DAD5(h_0)	DAD5	DAD4(X_4)	DAD5	A_1	These six MUX are driven identically by CMD1
DAD4(h_1)	DAD4	DAD3(X_5)	DAD4	A_2	
DAD3(h_2)	DAD3	DAD2(X_6)	DAD3	A_3	
DAD2(h_3)	DAD2	DAD1(X_7)	DAD2	A_4	
DAD1(h_4)	DAD1	DAD0(X_8)	DAD1	A_5	
DAD0(h_5)	DAD0	X_9	DAD0	A_6	
MSL3	PAGE	MSL3	PAGE	A_7	Driven by CMD2

PROM CODING PRINCIPLES

The PROM coding consists in the use of the 10 horizontal address bits (X_0, - - - -, X_9) to access the 640 pixels (organized in 64 segments of 10 pixels each).

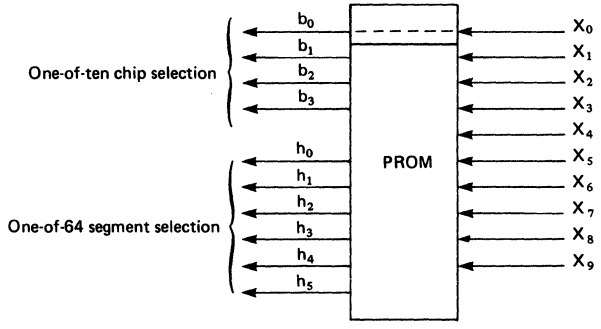
The 4 bits (b_0, b_1, b_2, b_3) are coding decimal numbers. Parity is maintained by BCD coding ; X_0 signal is therefore not coded inside the PROM and provides directly b_0.

Example : Considering the pixel with decimal abscissa X = 378 (17A in hexadecimal). This pixel is inside the 38th segment (h = 37 dec. or 25 hex.) with an abscissa x = 8.

The binary number 0101111010 (17A hex.) must be encoded into 1001011000 (258 hex.).

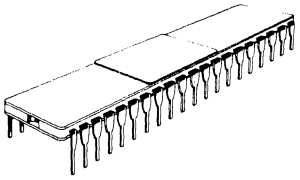
This principle allows transcoding of all horizontal address values. Transcoding must only be active (PROM selection) during write cycles ($\overline{ALL}$ = 1) when horizontal addresses are output ($\overline{RAS}$).

Note : This transcoding system may be adapted to other horizontal resolutions as 320, 384, 768. Horizontal resolutions are multiples of 64.

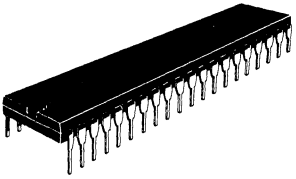


PHYSICAL DIMENSIONS

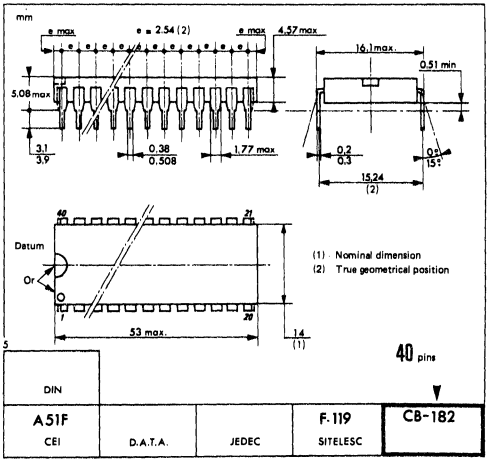
CB-182



C SUFFIX
CERAMIC PACKAGE



P SUFFIX
PLASTIC PACKAGE



THOMSON COMPONENTS MOSTEK

EF9369

SINGLE CHIP COLOR PALETTE

The EF9369 single chip palette provides a low cost, yet remarkable enhancement for any low to mid-range color graphics application. It allows displaying up to 16 different colors, each of these colors being freely selected out of 4096 preset values. EF9369 contains a 16 register color look-up table, three 4-bit D/A converters and a microprocessor interface for color loading.

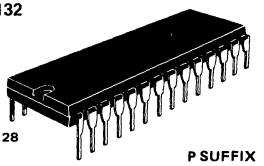
- On chip color look-up table
- 4096 color palette (16 colors selected from 4096)
- On-chip three 4-bit resolution video DACs with γ law correction
- Dot rate up to 30 Megadots per second
- Marking bit for inlay purpose
- Versatile microprocessor interface :
  - directly compatible with address/data multiplexed 8-bit microprocessor bus such as 6801, EF6805CT, 8051...
  - directly compatible with non-multiplexed 8 or 16-bit microprocessor bus (6809, 6502, 68008...).
- Single 5 V supply
- HMOS 2 technology.

HMOS2

SINGLE CHIP COLOR PALETTE

CASES

CB-132

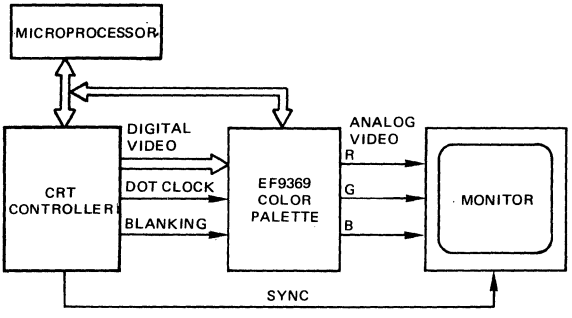


P SUFFIX
PLASTIC PACKAGE

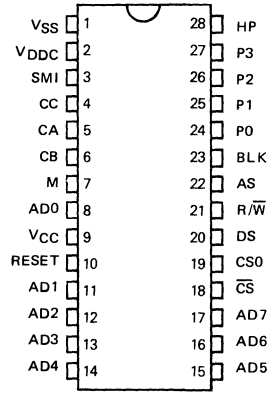
CB-520

FN SUFFIX
PLCC 28

TYPICAL APPLICATION



PIN ASSIGNMENT



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply voltage	V_{CC}*	-0.3 to 7.0	V
Input voltage	V_{in}*	-0.3 to 7.0	V
Operating temperature range	T_A	0 to 70	°C
Storage temperature range	T_{stg}	-55 to 150	°C
Max power dissipation	P_{Dm}	0.45	W

*With respect to V_{SS}

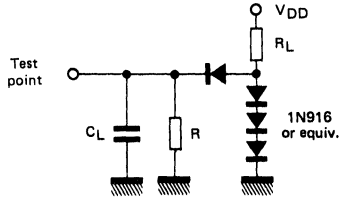
Stresses above those hereby listed may cause permanent damage to the device. The ratings are stress ones only and functional operation of the device at these or any conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. Standard MOS circuits handling procedure should be used to avoid possible damage to the device.

ELECTRICAL OPERATING CHARACTERISTICS

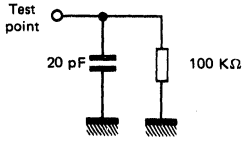
(V_{CC} = 5.0 V ± 5 %, V_{SS} = 0, T_A = 0 to 70°C))

Characteristic	Symbol	Min	Typ	Max	Unit
Supply voltage	V_{CC}	4.75	5	5.25	V
Analog supply voltage	V_{DDC}	–	V_{CC}	TBD	V
Analog supply current	I_{DDC}	–	20	–	mA
Input low voltage	V_{IL}	-0.3	–	0.8	V
Input high voltage RESET	V_{IH}	3	–	V_{CC}	V
Input high voltage All other inputs		2	–	V_{CC}	
Input leakage current	I_{in}	–	–	20	μA
Output high voltage (I_{load} = -500 μA)	V_{OH}	2.4	–	–	V
Output low voltage (I_{load} = 1.6 mA)	V_{OL}	–	–	0.4	V
Power dissipation	P_D	–	250	–	mW
Input capacitance	C_{in}	–	–	15	pF
Three state (off state) input current	I_{TSI}	–	–	10	μA

Test load for digital output



Test load for analog output



	AD(0:7)	M
C	100 pF	50 pF
R_L	1 KΩ	3.3 KΩ
R	4.7 KΩ	4.7 KΩ

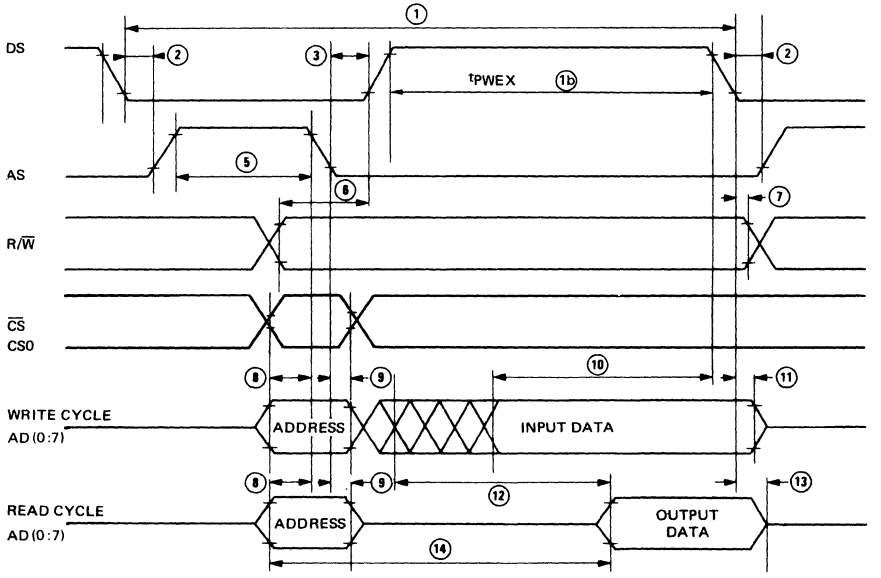
MICROPROCESSOR INTERFACE TIMING AD(0:7), AS, DS, R/$\overline{W}$, $\overline{CS}$, CS0

V_{CC} = 5.0 ± 5 %, T_A = 0° to + 70°C, C_L = 100 pF on AD(0:7)

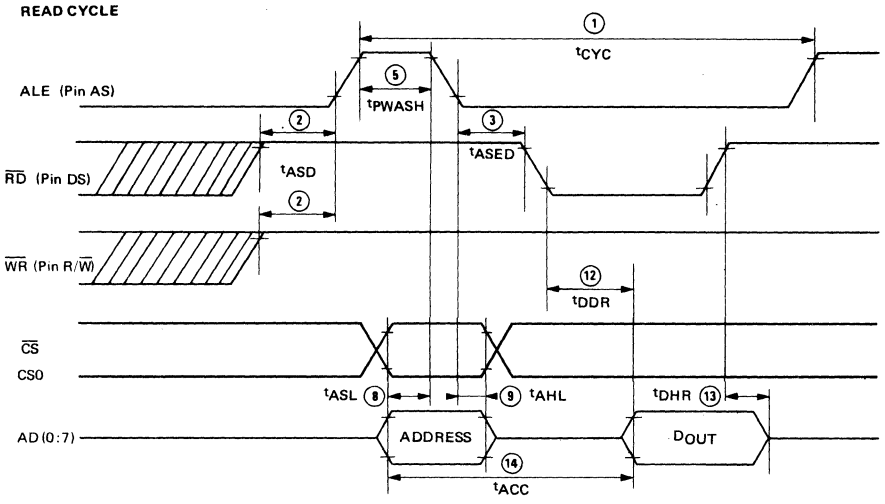
Reference levels : V_{IL} = 0.8 V and V_{IH} = 2 V on all inputs ; V_{OL} = 0.4 V and V_{OH} = 2.4 V on all outputs.

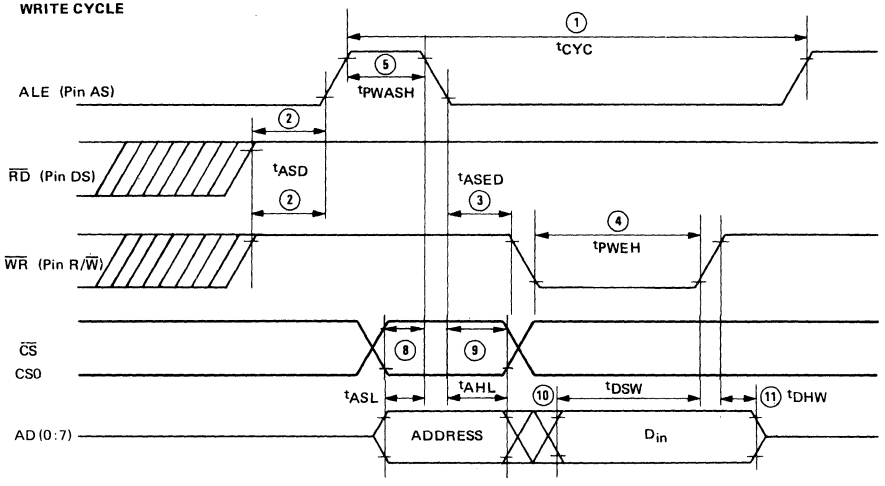
Ident. number	Characteristic	Symbol	Min	Typ	Max	Unit
1	Cycle time	t_{CYC}	400	–	–	ns
1b	DS pulse width high time	t_{PWEX}	200	–	–	ns
1c	DS pulse width low time (Timing 3)	t_{PWEL}	100	–	10000	ns
2	DS low to AS high (timing 1) DS high or R/$\overline{W}$ high to AS high (timing 2)	t_{ASD}	30	–	–	ns
3	AS low to DS high (timing 1) AS low to DS low or R/$\overline{W}$ low (timing 2)	t_{ASED}	30	–	–	ns
4	Write pulse width	t_{PWEH}	200	–	–	ns
5	AS pulse width	t_{PWASH}	100	–	–	ns
6	R/$\overline{W}$ to DS setup time (timing 1)	t_{RWS}	100	–	–	ns
6b	R/$\overline{W}$, AS, $\overline{CS}$, CS0 to DS setup time (timing 3)		100	–	–	
7	R/$\overline{W}$ to DS hold time (timing 1)	t_{RWH}	10	–	–	ns
8	Address and $\overline{CS}$, CS0 setup time	t_{ASL}	20	–	–	ns
9	Address and $\overline{CS}$, CS0 hold time	t_{AHL}	20	–	–	ns
10	Data setup time (write cycle)	t_{DSW}	100	–	–	ns
11	Data hold time (write cycle)	t_{DHW}	10	–	–	ns
12	Data access time from DS (read cycle)	t_{DDR}	–	–	150	ns
13	DS inactive to high impedance state time (read cycle)	t_{DHR}	10	–	80	ns
14	Address to data valid access time	t_{ACC}	–	–	300	ns

TIMING DIAGRAM 1 – MULTIPLEXED MODE - MOTOROLA TYPE (SMI = V_{SS})

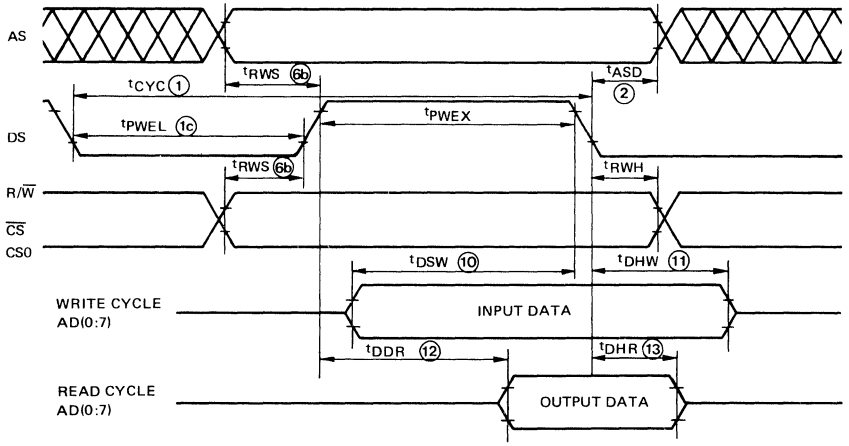


TIMING DIAGRAM 2 – MULTIPLEXED MODE - INTEL TYPE (SMI = V_{SS})





TIMING DIAGRAM 3 – NON-MULTIPLEXED MODE (SMI = V_{CC})



DIGITAL VIDEO SIGNALS - HP, P(0:3), BLK, M, RESET

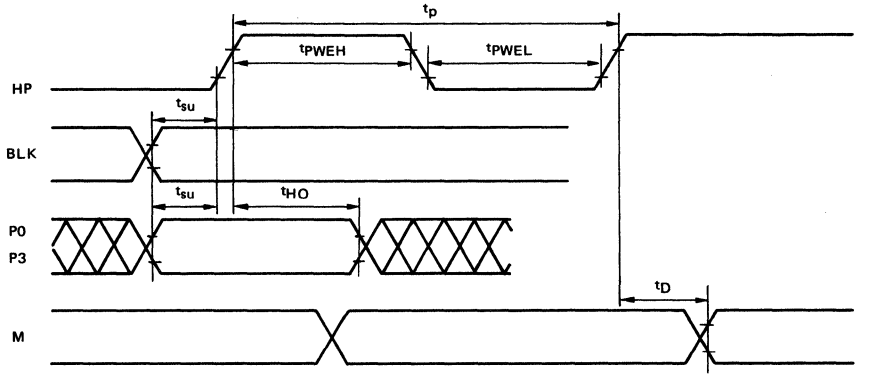
V_{CC} = 5.0 ± 5 %, T_A = 0°C to + 70°C, C_L = 50 pF on M.

Reference levels : V_{IL} = 0.8 V and V_{IH} = 2 V on all inputs ;

V_{OL} = 0.4 V and V_{OH} = 2.4 V on all outputs.

Characteristic	Symbol	EF9369		EF9369-30		Unit
		Min	Max	Min	Max	
HP clock period	t_P	58	1000	33	1000	ns
HP high pulse width	t_{PWEH}	25	–	13	–	ns
HP low pulse width	t_{PWEL}	25	–	13	–	ns
BLK and P(0:3) set up time to HP	t_{SU}	5	–	5	–	ns
BLK and P(0:3) hold time from HP	t_{HO}	10	–	10	–	ns
M output delay from HP	t_D	–	45	–	30	ns
RESET high pulse width	t_{PWRL}	400	–	400	–	ns

TIMING DIAGRAM 4



ANALOG VIDEO OUTPUTS - CA, CB, CC

V_{DCC} = 5 V, T_A = 0° to+ 70°C, C_L = 20 pF, R_L = 100 kΩ

TABLE 1

Binary input		Analog output (V)		
		Min	Typ	Max
Low level	0000	–	0.8	–
	0001	–	1.18	–
	0010	–	1.28	–
	0011	–	1.36	–
	0100	–	1.42	–
	0101	–	1.47	–
	0110	–	1.52	–
	0111	–	1.56	–
	1000	–	1.60	–
	1001	–	1.63	–
	1010	–	1.66	–
	1011	–	1.69	–
	1100	–	1.72	–
	1101	–	1.75	–
	1110	–	1.78	–
High level	1111	–	1.80	–

Note :

The internal A/D converters deliver on CA, CB and CC outputs 16 levels with γ law correction ($\gamma = 2.8$). The typical transfer characteristic is given by :

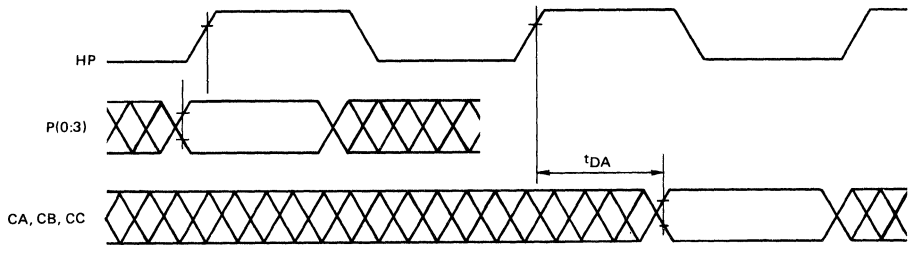
$$V = \left(\frac{N}{15}\right)^{\frac{1}{2.8}} \cdot \frac{V_{DDC}}{5} + 0.16\, V_{DCC}$$

Where N is the binary input value.

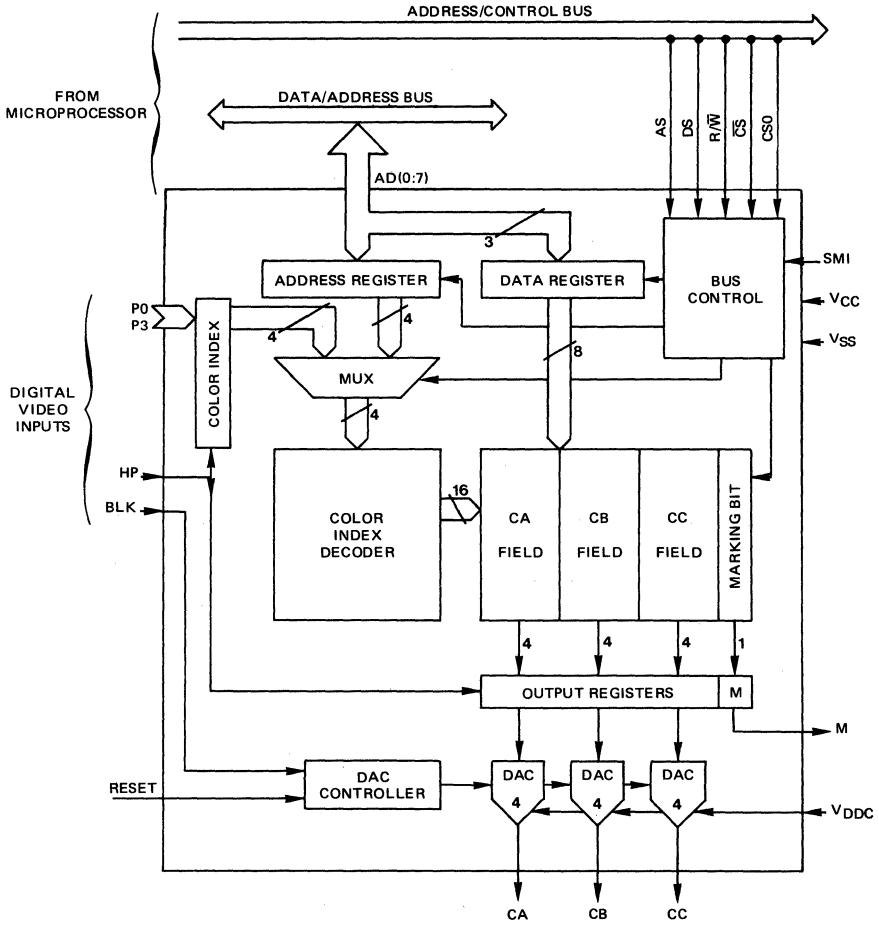
The typical analog video output impedance is 300 Ω for EF9369-30 and 400 Ω for EF9369.

Characteristic	Symbol	Min	Typ	Max	Unit
CA, CB, CC outputs from HP	t_{DA}	–	80	–	ns

TIMING DIAGRAM 5



BLOCK DIAGRAM



PIN DESCRIPTION

MICROPROCESSOR INTERFACE

All the input/output pins are TTL compatible.

NAME	PIN TYPE	N°	FUNCTION	DESCRIPTION
AD(0:7)	I/O	8-11-14 15 - 17	Multiplexed address/data bus	These 8 bidirectional pins are to be connected to the microprocessor system bus.
SMI	I	3	Interface mode select	When this input is connected to V_{CC}, the EF9369 is in the non multiplexed mode. When this input is connected to V_{SS} (ground), the EF9369 is in a multiplexed mode to provide a direct interface with either Motorola or Intel type microprocessor.
AS	I	22	Address strobe	In non-multiplexed mode, this input selects either the address register (AS = 1) or the data register (AS = 0) to be accessed. In multiplexed mode, the falling edge of this control signal latches the address on the AD(0:7) lines, the state of the Data Strobe (DS) and Chip Select lines ($\overline{CS}$, CS0). When using Intel type microprocessor, this input must be connected to the ALE control line.
DS	I	20	Data strobe	In non-multiplexed mode, this active high control signal enables the AD(0:7) input/output buffers and strobes data to/from the EF9369. This signal is usually derived from the processor E ($\phi 2$) clock. In multiplexed mode, the input is strobed by the falling edge of AS. The strobe value selects either Motorola or Intel type. When using an Intel type microprocessor, DS must be connected to the $\overline{RD}$ control line. With a Motorola type microprocessor, DS must be connected to E($\phi 2$) clock.
R/$\overline{W}$	I	21	Read/Write	This control signal determines whether the EF9369 is read (R/$\overline{W}$ = 1) or written (R/$\overline{W}$ = 0). When using Intel type microprocessor, this input must be connected to the $\overline{WR}$ control line.
$\overline{CS}$ CS0	I	18 19	Chip Select	$\overline{CS}$ must be low and CS0 must be high to select the EF9369. In non-multiplexed mode, the EF9369 remains selected as long as the selection condition is met. In multiplexed mode, the selection condition is latched when AS is low.

VIDEO INTERFACE

NAME	PIN TYPE	N°	FUNCTION	DESCRIPTION
P(0:3)	I	24 - 27	Pixel inputs	These four TTL compatible inputs are strobed by HP into the color index register to address the color look-up table.
HP	I	28	Dot clock	The rising edge of this input latches the P(0:3) and BLK inputs into the EF9369 and the data out of the color look-up table into the output registers.
M	O	7	Marking	This output is synchronised by HP and delivers the marking bit value from the color look-up table.
CA CB CC	O	5 6 4	Color outputs	These three analog high impedance outputs deliver the color signal levels from the internal D/A converters (DAC). The delay between CA, CB, CC outputs and the latched value P(0:3) is one HP clock period plus t_{DA} (see Timing Diagram 5).
BLK	I	23	Blanking	A high level on this input forces the CA, CB, CC and M outputs to low level.
RESET	I	10	Reset	This active high input forces the CA, CB, CC, outputs to low level until the next microprocessor access to the device.

OTHER PINS

V_{CC}	S	9	Power supply	+ 5 V.
V_{DDC}	S	2	Analog power supply	Power supply for the internal DACs. This input can be connected to V_{CC}.
V_{SS}	S	1	Power supply	Ground.

FUNCTIONAL DESCRIPTION

EF9369 contains a 16 register Color-Look Up Table (CLUT). Each of these 13-bit register holds three 4-bit color fields CA (0:3), CB (0:3) and CC (0:3) and a marking bit M.

These registers can be accessed (read or write) by the microprocessor through the microprocessor interface. These registers are also read by the video process : a 4-bit pixel value and a clock must be provided at pixel rate to the P(0:3) and HP input pins. These signals may be delivered either by 4 video shift registers and the shifting clock of a bit map CRT controller or by an alphanumeric or semigraphic CRT controller. The pixel value, after clock resynchronisation, is used as a color index : it selects one out of the 16 CLUT registers. Each color field of the selected register is converted to an analog signal and delivered to one of the CA, CB or CC output. The marking bit is directly routed to the M output. When the CA, CB and CC outputs are used as RGB analog signals, one color out of 4096 is associated to each pixel value. In short this process freely maps a 16 color index set into a 4096 color set.

MICROPROCESSOR INTERFACE

The 8-bit microprocessor interface gives access (read or write) to the CLUT which is addressed as a 32 byte table. The 13-bit color register # N (N = 0 to 15) is accessed at address 2N and 2N + 1. Even address holds CA (0:3) and CB (0:3), odd address holds CC (0:3) and M (see fig. 1).

EF9369 provides two bus modes through the SMI programming pin :

- Multiplexed mode for address/data multiplexed 8-bit microprocessor bus.
- Non-multiplexed mode for non-multiplexed 8 or 16-bit microprocessor bus.

MULTIPLEXED MODE (SMI connected to V_{SS})

In this mode, EF9369 can be directly connected to popular address/data multiplexed microprocessor, either Motorola type (6801, EF6805CT...) or Intel type (8048, 8051, 8088..). In this last case the EF9369 AS, DS and R/$\overline{W}$ inputs must be connected respectively to the ALE, $\overline{RD}$ and $\overline{WR}$ microprocessor control lines.

FIGURE 1 – CLUT ADDRESSING

Color Look-Up Table (CLUT)								CLUT Byte Address								Register Index
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	#
CB3	CB2	CB1	CB0	CA3	CA2	CA1	CA0	X	X	X	0	0	0	0	0	0
X	X	X	M	CC3	CC2	CC1	CC0	X	X	X	0	0	0	0	1	
CB3	CB2	CB1	CB0	CA3	CA2	CA1	CA0	X	X	X	0	0	0	1	0	1
X	X	X	M	CC3	CC2	CC1	CC0	X	X	X	0	0	0	1	1	
≈																≈
CB3	CB2	CB1	CB0	CA3	CA2	CA1	CA0	X	X	X	1	1	1	1	0	15
X	X	X	M	CC3	CC2	CC1	CC0	X	X	X	1	1	1	1	1	

X = don't care.

In this mode, EF9369 maps into the microprocessor addressing space as 32 CLUT byte address. Random access to one byte takes one cycle : on the falling edge of the AS input, EF9369 latches AD (0:7) into the on-chip address register, the DS and chip select lines into dedicated flip-flops. The strobed value of DS allows recognition of Intel or Motorola type for further processing. (See pin description section and microprocessor timing diagrams for details). When the EF9369 chip select lines enable selection, the addressed byte is accessed during the data phase of the cycle.

NON MULTIPLEXED MODE (SMI connected to V_{CC})

In this mode EF9369 can be directly connected to any 8 or 16-bit, non multiplexed, microprocessor bus (6800, 6809, 6502, 68008...).

This mode provides an indirect, auto-incremented addressing scheme. EF9369 maps into the microprocessor addressing space as 2 byte address only. AS is used to select one out of 2 registers :

- the write only address register (5 bits) addressed when AS = 1.
- the read/write data register (8 bits) addressed when AS = 0.

Random access to a CLUT byte takes two bus cycles :
1/ Load the CLUT address into the address register.
2/ Access (read or write) the value in the data register.

After each access to the data register, the address register is automatically incremented modulo 32. This scheme allows sequential addressing to the CLUT without address reloading, the complete CLUT can so be reloaded in 33 bus cycles.

VIDEO PROCESS

The CRT controller sends to EF9369 a pixel value on pins P (0:3), a pixel rate clock on HP input and a blanking signal on pin BLK. The pixel value is latched into the color index register by the rising edge of HP. The color index register selects one register in the CLUT. The color fields of the selected register are routed to 3 DACs and M is directly routed to the M digital output.

After impedance matching, the CA, CB, and CC outputs can be used to drive a RGB analog color monitor. Alternatively one of these outputs can be used to drive a monochrome monitor thus providing up to 16 gray levels. The marking digital output can be used to drive analog video switches, thus providing video overlay facility on a color per color basis.

The blanking input forces the analog outputs and the M output to low level thus allowing the beam to be switched off during retrace intervals.

NOTES

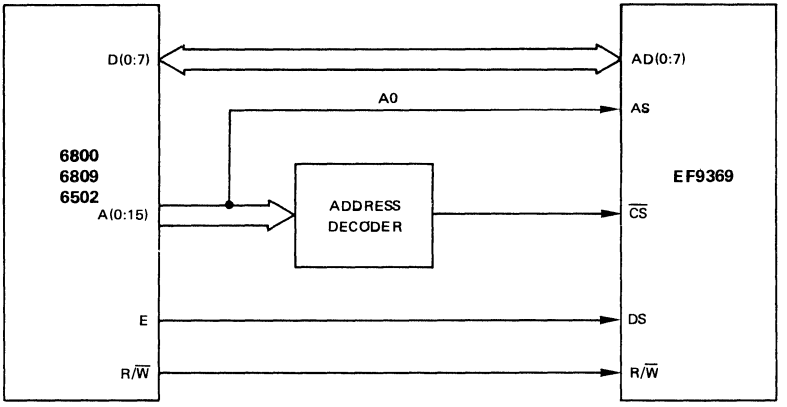
1. Each 4 bit-D/A converter is γ corrected in order to linearize the luminance driven on the screen versus the digital value. The typical digital to voltage conversion law is given table 1. The output voltages are proportionnal to the analog supply voltage V_{DDC}. When required, setting V_{DCC} allows a gain adjustment. But in most applications, V_{DDC} and V_{DD} can be derived from the same supply through independent decoupling.

2. CA, CB and CC are high impedance outputs (500 Ω typical) which require proper adaptation in most applications. THOMSON SEMICONDUCTORS TEA 5114 provides such a 1 V - 75 Ω low cost adaptation. (See fig. 2).

3. As the CLUT is shared between microprocessor access and video access, a low level is forced on the CA, CB, CC and M outputs during any chip select periods. To avoid to spoil the screen with black strokes it is recommended to access the CLUT from the microprocessor only during the retrace periods.

4. RESET - This input forces CA, CB, CC and M outputs to a low level until the next microprocessor access. At power on or at the beginning of a session RESET allows to keep a clean black screen until proper initialization.

INTERFACE WITH 6800/6809 TYPE MICROPROCESSOR



INTERFACE WITH EF6805CT

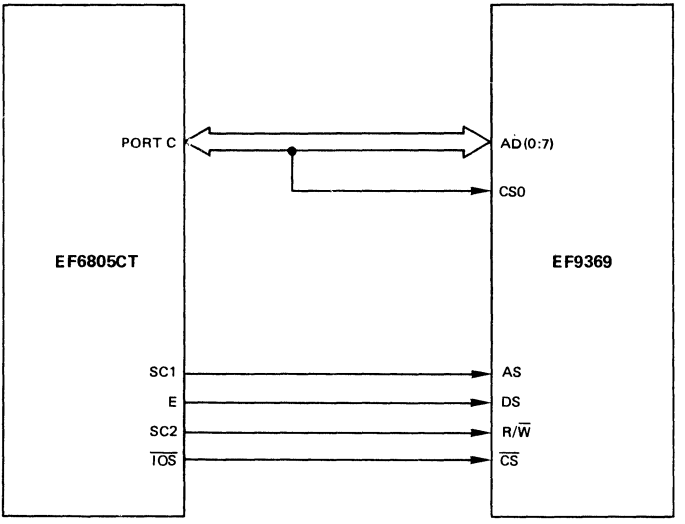
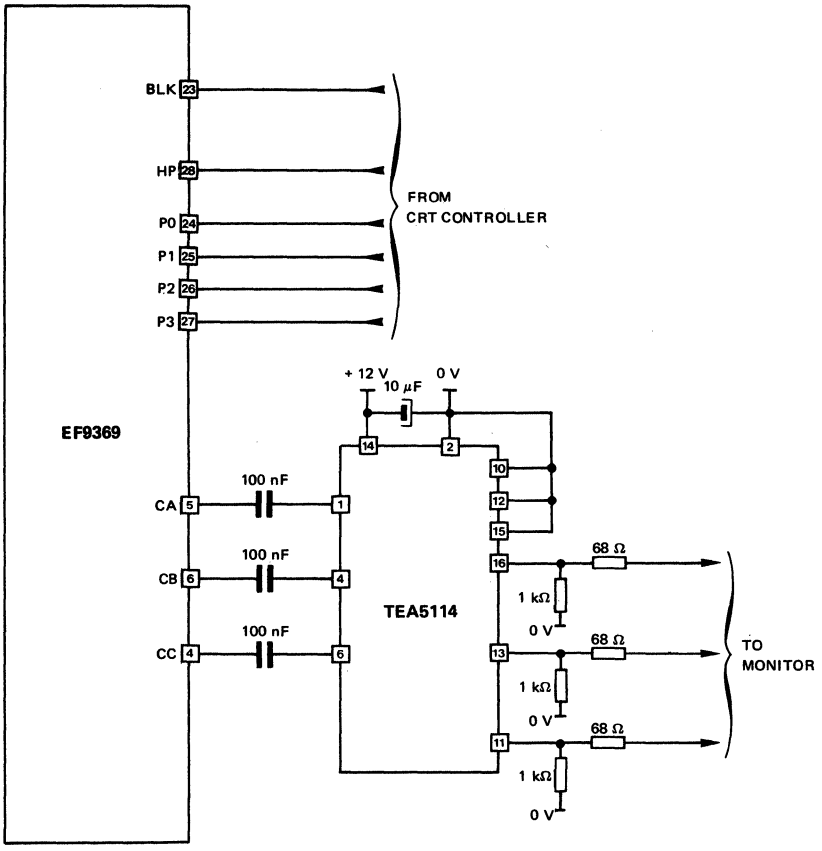
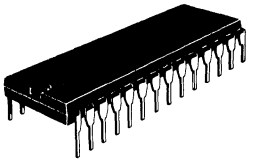


FIGURE 2 – TYPICAL 1 V - 75 Ω VIDEO INTERFACE



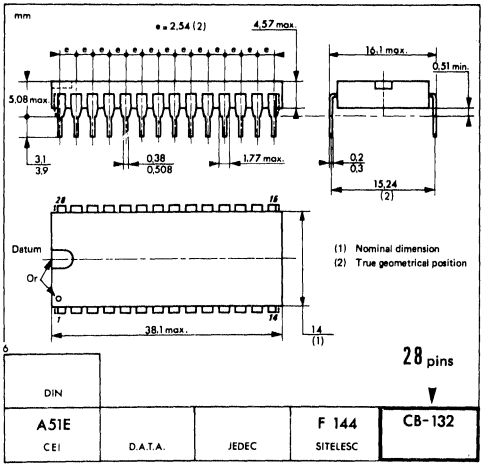
NOTE : Each digital or analog ground must be separately connected to EF9369 pin 1. (See AN-059).

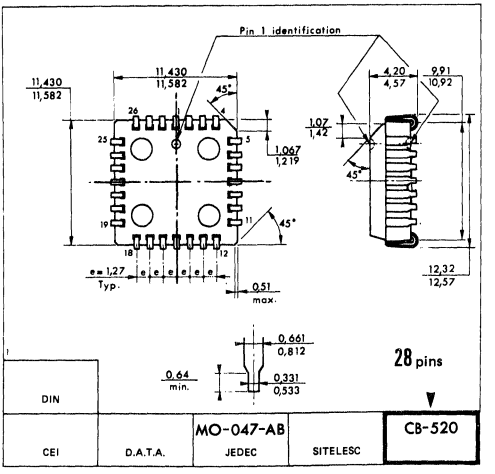
CB-132 **PHYSICAL DIMENSIONS** CB-520

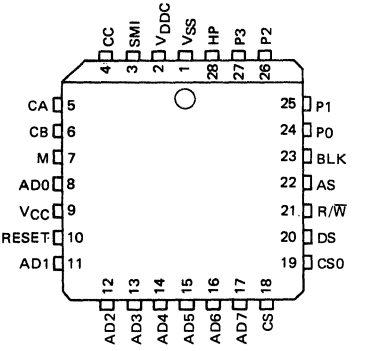


P SUFFIX
PLASTIC PACKAGE

FN SUFFIX
PLCC 28







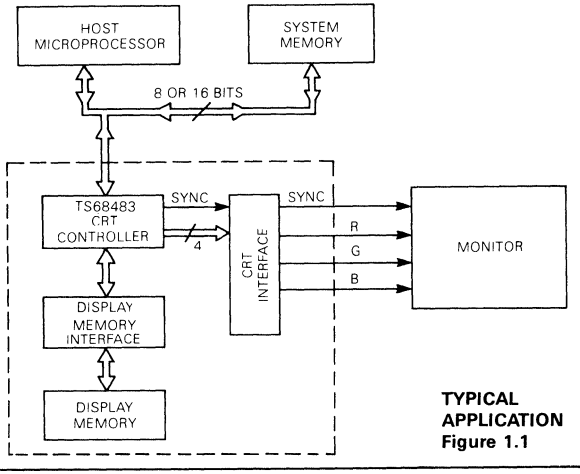
TS68483

ADVANCED GRAPHIC AND ALPHANUMERIC CONTROLLER

ADVANCE INFORMATION

The TS68483 is an advanced color graphic processor that drastically reduces the CPU software overhead for all graphic tasks in medium and high range graphic applications such as business and personal computer, industrial monitoring system and CAD systems.

- Fully programmable timing generator.
- Alphanumeric and graphic drawing capability.
- Easy to use and powerful command set:
  - VECTOR, ARC, CIRCLE with dot or pen concept and programmable line style,
  - Flexible area fill command with tiling pattern,
  - Very fast block move operation,
  - Character drawing command, any size and fonts available.
- Large frame buffer addressing space (8 megabyte) up to 16 planes of 2048 x 2048.
- Up to 256 color capabilities.
- Mask bit planes for general clipping purpose.
- Frame buffer can be built with standard 64K or 256K DRAM or Dual-Port-Memories (Video-RAM).
- External Synchronization capability.
- On chip video shift registers for Dot rate less than 15 Megadots/s.
- 8 or 16-bit bus interface compatible with market standard microprocessors.
- HMOS 2 technology.
- 68 - pin PLCC package.



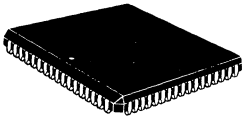
TYPICAL APPLICATION
Figure 1.1

HMOS2

ADVANCED GRAPHIC AND ALPHANUMERIC CONTROLLER

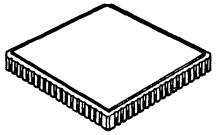
CASES

CB-523



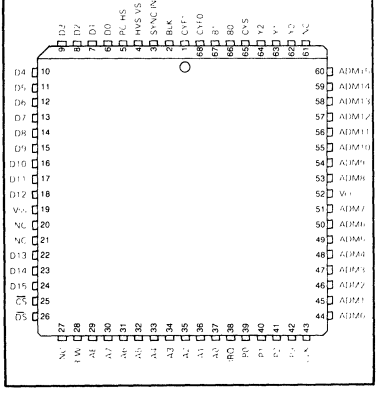
FN SUFFIX
PLCC 68

CB-710



E SUFFIX
LCCC 68

PIN ASSIGNMENT



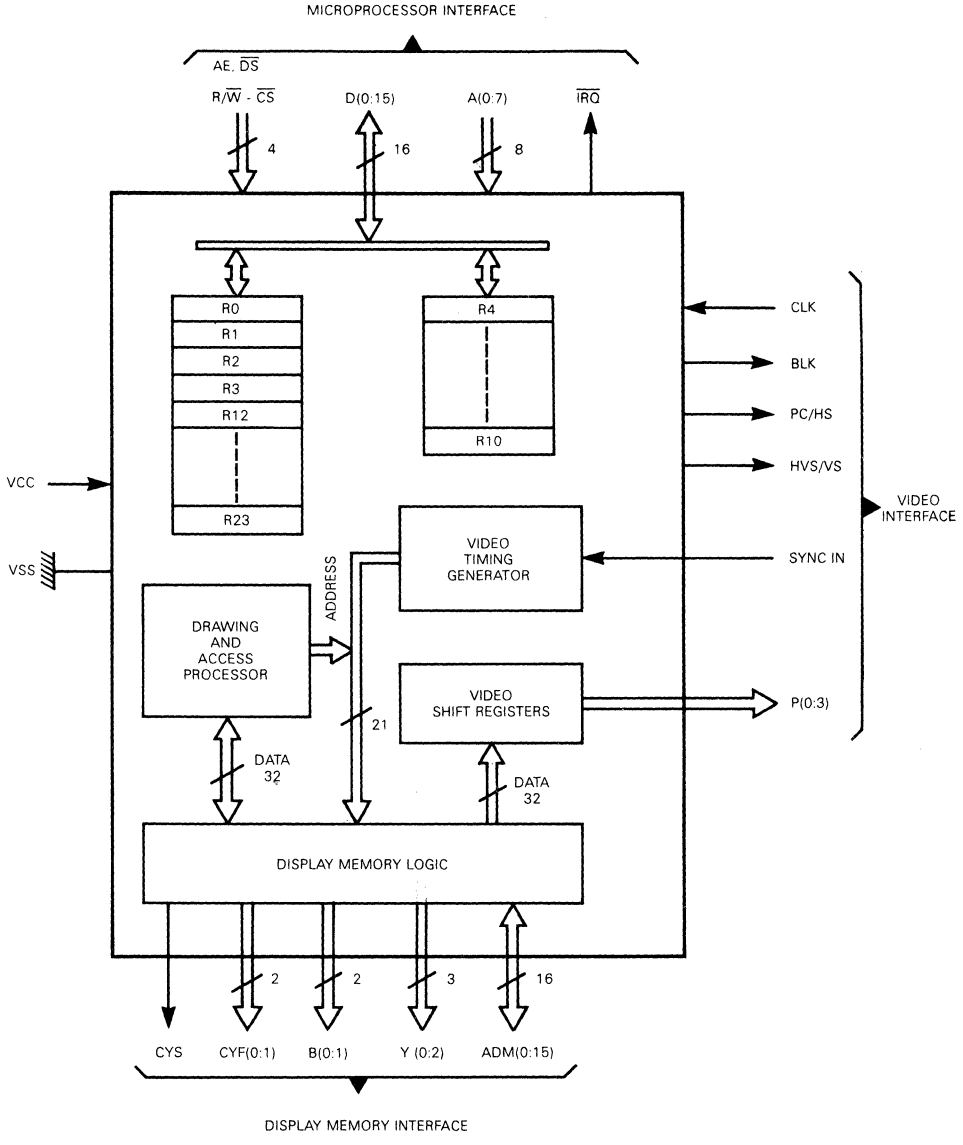
SECTION 1
GENERAL OPERATION

1.1 INTRODUCTION

THOMSON SEMICONDUCTEURS TS68483 is an advanced color graphics controller chip. It is directly compatible with most popular 8 or 16-bit microprocessors.

Its display memory, containing the frame buffer and the character generators, may be assembled from standard dynamic RAM components.

On-chip video shift registers and fully programmable Video Timing Generator allow the TS68483 to be used in a wide range of terminals or computer design.



TS68483 BLOCK DIAGRAM

TS68483 PIN DESCRIPTION

MICROPROCESSOR INTERFACE

Name	PIN Type	N°	Function	Description
D(0:15)	I/O	6-18 22-24	Data bus	These sixteen bidirectional pins provide communication with either an 8 or 16-bit host microprocessor data bus.
A(0:7)	I	37-30	Address bus	These eight pins select the internal register to be accessed. The address can be latched by AE for direct connection to address/data multiplexed microprocessor busses.
AE	I	29	Address Enable	When TS68483 is connected to a non-multiplexed microprocessor bus, this input must be wired to VCC. For direct connection to a multiplexed microprocessor bus, the falling edge of AE latches the address on A (0:7) pins and the $\overline{CS}$ input. With an Intel type microprocessor AE is connected to the processor Address Latch Enable (ALE) signal.
$\overline{DS}$	I	26	Data strobe	Active low – In non-multiplexed bus mode, $\overline{DS}$ low enables the bidirectionnal data buffers and latches the A(0:7) lines on its high to low transition. Data to be written are latched on the rising edge of this signal. – In multiplexed bus mode, this signal low enables the output data buffers during a read cycle. With Intel microprocessors, this pin is connected to the RD signal.
R/$\overline{W}$	I	28	Read/Write	– In non-multiplexed bus mode, this signal controls the direction of data flow through the bidirectional data buffers. – In multiplexed bus mode, this signal low enables the input data buffers. The entering data are latched on its rising edge. With Intel microprocessors, this pin is connected to the $\overline{WR}$ signal.
$\overline{CS}$	I	25	Chip select	This input selects the TS68483 registers for the current bus cycle. A low level corresponds to an asserted chip select. In multiplexed mode, this input is strobed by AE.
$\overline{IRQ}$	O	38	Interrupt Request	This active-low open drain output acts to interrupt the microprocessor.

MEMORY INTERFACE

Name	PIN Type	N°	Function	Description
ADM(0:15)	I/O	44-51 53-60	Address/Data Memory	These multiplexed pins act as address and data bus for display memory interface.
CYS	O	65	Memory Cycle Start	The falling edge of this output indicates the beginning of a memory cycle.
Y(0:2)	O	62-64	Memory Address	These outputs provide the least significant bits of the Y logical address.
B(0:1)	O	66-67	Bank number	These outputs provide the number of the memory bank to be accessed during the current memory cycle.
CYF(0:1)	O	68,1	Memory Cycle Status	These outputs indicate the nature of the current memory cycle (Read, Write, Refresh, Display).

VIDEO INTERFACE

Name	PIN Type	N°	Function	Description
P(0:3)	O	39-42	Video Shift Register Outputs	These four pins correspond to the outputs of the internal video shift registers.
PC/HS	O	5	Phase comparator/ Horizontal sync.	This output can be programmed to provide either the phase comparator output or the horizontal sync. signal.
HVS/VS	O	4	Composite or Vertical sync.	This output can be programmed to provide either the composite sync. signal or the vertical sync. signal.
SYNC IN	I	3	External Sync Input	This input receives an external composite sync. signal to synchronize TS68483. This input must ge grounded if not used.
BLK	O	2	Blanking	This output provides the blanking interval information.

OTHER PINS

Name	PIN Type	N°	Function	Description
VCC	S	52	Power supply	+ 5 V supply
VSS	S	19	Ground	Ground
CLK	I	43	Clock	Clock input

1.2 TYPICAL APPLICATION BUILDING BLOCKS (See figure 1.1.)

In a typical application using TS68483, a host processor drives a display unit which drives in turn a color CRT monitor.

The display unit consists of four hardware building blocks:

- an TS68483 advanced graphics controller,
- a display memory (dynamic RAM),
- a display memory interface, comprising a few TTL parts,
- a CRT interface or CRT drivers.

For enhanced graphics, the CRT interface may include a color look-up table circuit such as EF9369. For high pixel rate (over 15 Mpixels/s), the CRT interface must include high speed video shift registers.

The display memory interface and organization are discussed in full details in the User's Manual.

1.3 TS68483 FUNCTIONS

All the TS68483 functions are under the control of the host microprocessor via 24 directly accessible 16-bit registers. These registers are referred to by their decimal index (R0 - R23). See figure 1.2

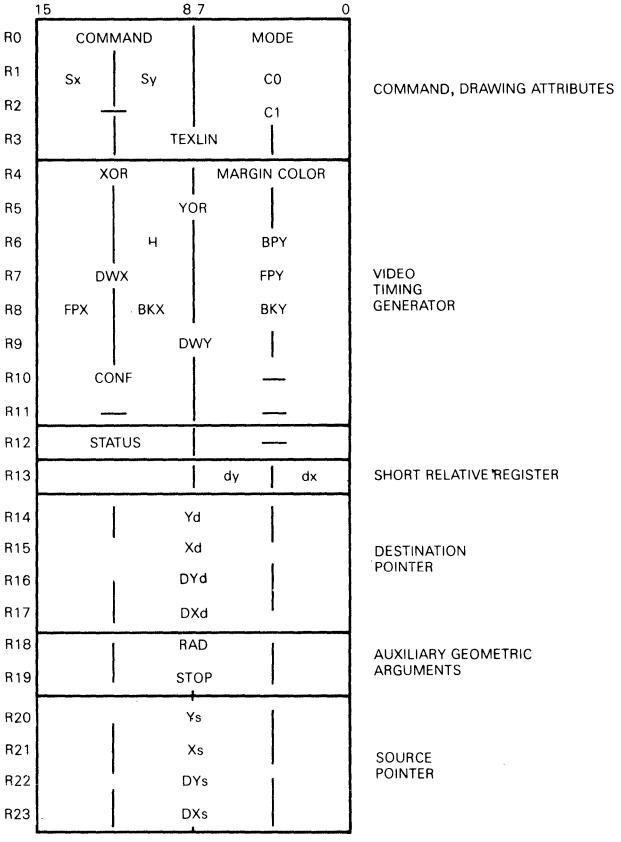


FIGURE 1.2. - REGISTER MAP

1. Video timing and display processor (R4 to R10)
The video timing generator is fully programmable: any popular horizontal scanning period from 20 µs to 64 µs may be freely combined with any number of lines per field (up to 1024). The address of the display viewport (this part of the display memory to be actually displayed on the screen) is fully programmable. The display processor provides the display dynamic RAM refresh (see video timing generator section for details).

2. Drawing and access commands (R0 to R3, R12 to R23)
The 16 remaining registers are used to specify a comprehensive set of commands. The highly orthogonal drawing command set allows the user to "draw" in the display memory such basic patterns as lines, arcs, polylines, polyarcs, rectangles and characters. Efficient procedures are available for either area filling and tiling or line drawing and texturing. Lines may be drawn with a PEN in order to get thick strokes. Any drawing is specified in a $2^{13} \times 2^{13}$ drawing coordinate system.

To access the display memory, the host microprocessor has an indirect, sequential access to any "window". Access commands can be used to load the character generators as well as to load or save arbitrary windows stored in the frame buffer.

1.4 DATA TYPE DEFINITIONS

PIXEL: this is the smallest color spot displayable on the CRT.

PEL: a Picture Element is the coding of a PIXEL in the display memory. The TS68483 can handle 4 different PEL formats:

- 4 color bits – short
- 4 color bits + 1 mask bit – short masked
- 8 color bits – long
- 8 color bits + 1 mask bit – long masked

DRAWING COORDINATES: (See figure 1.3.)
The drawing commands are specified and computed in a $2^{13} \times 2^{13}$ cyclical coordinate system. The drawing coordinates are clipped and mapped into the $2^{11} \times 2^{11}$ display memory addressing space. Further clipping to the actual frame buffer size may be performed by the user designed memory interface.

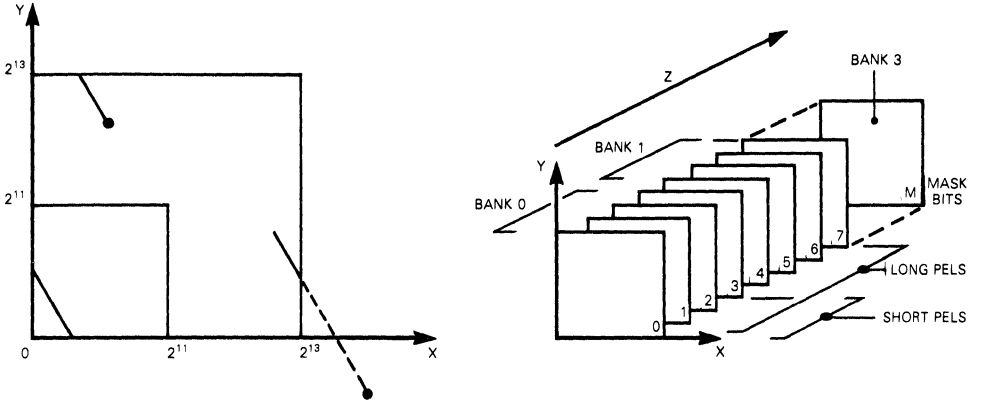


FIGURE 1.3. - CYCLICAL DRAWING COORDINATES TO DISPLAY MEMORY MAPPING

DISPLAY MEMORY:
This is the private memory dedicated to the display unit. This memory is addressed as four banks of 4-bit plane each.

BIT PLANE:
Each bit plane has a maximum capacity of $2^{11} \times 2^{11}$ bits. A byte wide organization of each bit plane is required.

MEMORY ADDRESS: (See figure 1.4.)
In order to address one bit in the display memory, the user must specify:

- A bank number (2 bits) B = 0 to 3
- A bit plane number (2 bits) Z = 0 to 3
- A Y address (11 bits) Y = 0 to 2047
- An X address (11 bits) X = 0 to 2047

MEMORY WORD: (See figure 1.4.)
A 32-bit memory word can be either read or written during each memory cycle (8 CLK periods), one byte at a time in each bit plane in the addressed bank. The memory bandwidth is in the 6 to 8 Mbytes/s range.

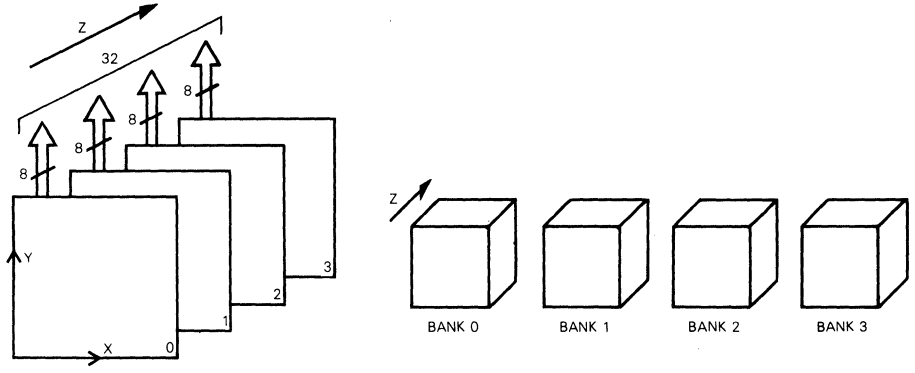


FIGURE 1.4. - THE DISPLAY MEMORY ADDRESSING SPACE

VIEWPORT:
This is any rectangular array of pels located in the display memory.

FRAME BUFFER:
This is the biggest viewport which can be held in the display memory. The frame buffer maps a window at the origin of the drawing coordinates. A short pel frame buffer may be located in any bank. A long pel frame buffer must be located in the "bank 0, bank 1" pair.

DISPLAY VIEWPORT:
This is the viewport which is displayed on screen.

MASK BIT PLANE:
When masked pels are used, a mask bit plane must be associated to a frame buffer. Mask bit planes may be located in any plane of bank 3.

CELL:
A CELL is any pattern stored in the display memory as a rectangular array of bit mapped elements. The drawing of any CELL may be specified with a scaling factor.

CHARACTER:
This is a one bit per element CELL. It may be stored in any bit plane, then colored and drawn in a frame buffer by use of PRINT CHARACTER command.

OBJECT:
This is a one short pel per element CELL. It may be drawn or loaded in a frame buffer. A source mask bit may be associated to each element. An OBJECT may then be printed in another location by use of a PRINT OBJECT command.

PEN:
This is the pattern which is repeatedly drawn along the coordinates defined by either a LINE or an ARC command. The PEN may be a DOT (single pel), a CHARACTER or an OBJECT.

SECTION 2
COMMANDS

2.1. INTRODUCTION

The command set is strongly organized in five subset or command types.

DRAWING COMMANDS:
- LINEAR (line, arc)
- AREA (rectangle, trapezium, polygon, polyarc)
- PRINT CELL (print character, print object)

ACCESS COMMANDS

CONTROL COMMANDS (move cursor, abort)

The commands are parametered; this means that any command can be executed with options freely selected out of a given option set. This option set is common for any command of a given type. For example, any drawing command may be parametered for destination mask bit use.

COMMAND SET STRUCTURE

Command	Drawing mode	Type	Group
Line Arc	Up to the pen	Linear	Drawing
Rectangle Trapezium Polygon Polyarc	Monochrome	Area	
Print char Print object	Bichrome Polychrome	Cell	
Load viewport Save viewport Modify viewport	–	Access	Management
Move cursor Abort	–	Control	

FIGURE 2.1. - COMMAND SET STRUCTURE

The command code also defines the command type and its parameters. A command is completely defined when a value has been set for each of its arguments.

These arguments are:

- the geometric arguments given in the drawing coordinate system for every drawing command. They are automatically mapped into the destination frame buffer;
- the parametric values are the values required by the selected parameters;
- the attribute values are the other values required by a drawing command; colors or scaling factors for example;
- the display memory addresses.

The command code is specified in register R0. Before initiating a command execution, each argument must be specified in its dedicated register: – an Xd, Yd drawing coordinate pair for example, is always located in registers R14, R15.

The monitoring of a command execution is done by reading the status register R12 or using the $\overline{IRQ}$ signal.

2.2. POINTERS AND GEOMETRIC ARGUMENTS

Pointers are used to specify main geometric arguments and display memory addresses.

2.2.1. Display memory address

A bit in the display memory is addressed by:

- a bank number B = 0 to 3
- a plane number Z = 0 to 3
- an X address X = 0 to 2047
- a Y address Y = 0 to 2047

2.2.2. Destination pointer: registers R14 to R17

This pointer gives the coordinate (Xd, Yd) and dimension (DXd, DYd) of either a line or a window in the drawing coordinate system. These drawing coordinates are easily mapped into a PEL DISPLAY MEMORY address.
(X, Y) coordinates are clipped to 11 bits in order to get the Xd, Yd destination pel addresses.

A bank number Bd must be explicitly provided to address a destination frame buffer. When long pels are used, Bd must be even.

When masked pels are used, the destination mask plane number Zd (implicitly in bank 3) must also be provided.

2.2.3. Source pointer: registers R20 to R23

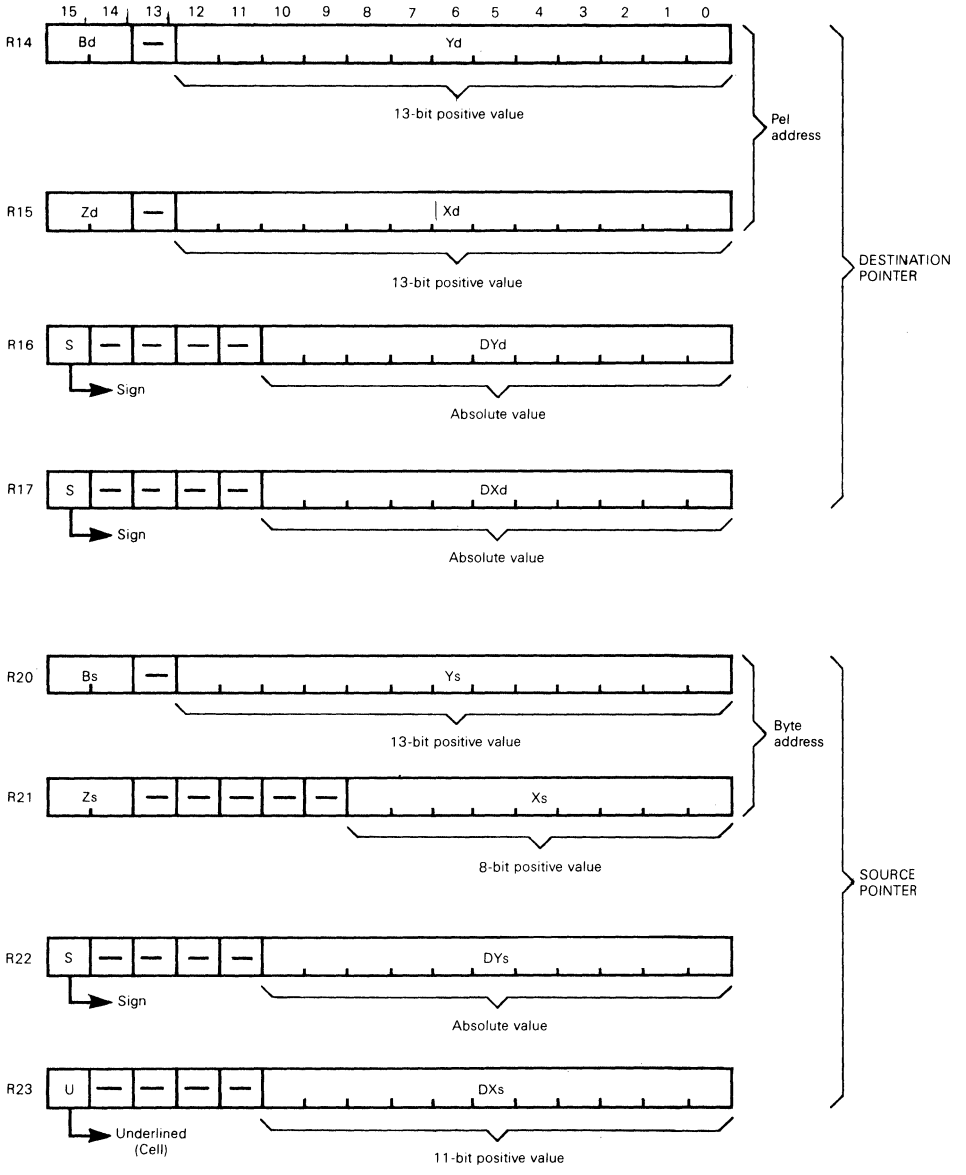
A source cell such as a character, a pen or an object, is addressed by the source pointer in the display memory.

A source pointer specifies:

- a bank number Bs = 0 to 3
- a Ys address Ys = 0 to 2047
- an Xs address; this address is a byte address so that the 3 LSBs are not specified Xs = 0 to 255
- a cell dimension DXs, DYs
- a bit plane address Zs.

When a character is addressed, Zs gives the plane number into the bank Bs. When an object is addressed Zs gives the source mask plane number in the bank B3.

FIGURE 2.2. - POINTERS



NOTE: Sign value: ● S = 0: positive
● S = 1: negative + absolute value.

2.2.4. Notes

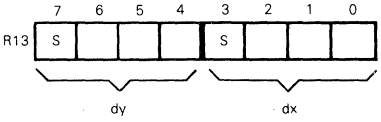
1. The TRAPEZIUM command makes a special use of R21. In this case, R21 holds an X1 drawing coordinate which has the same format as Xd.

2. The ARC and POLYARC commands require two extra geometric parameters (RAD and STOP). They are specified in the drawing coordinates system and stored in registers R18, R19.

3. Any drawing command may be parametered to use short incremental dimensions, DXY in register R13 instead of the standard DXd, DYd in the "R16, R17" register pair. (See figure 2.3).

4. The access commands use the destination pointer location as a data buffer. The memory addresses and dimension of the access viewport are then specified in the source pointer, independently of the data transfer.

5. DXd, DYd and DYs may specify a negative value. In this case, they must be coded by a sign (0 = positive, 1 = negative) and an 11-bit absolute value.



dx, dy = − 7 to + 7 (sign + absolute value)

FIGURE 2.3. - SHORT DIMENSION REGISTER R13

2.3. DESTINATION MASK AND SOURCE MASK

A mask bit may be associated to any pel stored in the display memory.

2.3.1. Destination mask use (DMU)

Any drawing command may be parametered for destination mask use. In this case, any destination pel cannot be modified when its mask bit is reset.

In other words:

When the destination mask use (DMU) parameter is set:
- a pel may be modified when its mask bit is set
- a pel cannot be modified when its mask bit is reset.

When the destination mask use (DMU) parameter is cleared:
- a pel may be modified, independently of its mask bit value.

This provides a very flexible clipping mechanism not restricted to rectangular windows. (See destination pointer section for destination mask bit addressing).

2.3.2. Source mask use (SMU)

A PRINT OBJECT command may be parametered for source mask use. In this case, the source mask bit associated with any source pel is read first. When its mask bit is cleared, a source pel is considered as transparent. (See source pointer section for source mask bit addressing).

In other words:

When the SMU parameter is set, the color of a destination pel, mapped by a given source pel, may take this source color value only when this source bit mask is set. The destination pel keeps its own color value when the source bit mask is cleared.

When the SMU parameter is cleared, a source pel color may be mapped into destination pel color independently of the source bit mask value.

The source bit mask acts as a TRANSPARENCY/OPACITY flag which is enabled by SMU. A PRINT OBJECT command may be independently parametered by both SMU and DMU. This provides a very powerful tiling, print object or move mechanism.

2.4. DRAWING ATTRIBUTES

The general drawing attributes are the colors, the drawing mode, and the scaling factor.

2.4.1. Colors: registers R1 and R2 (See figure 2.4.)

Two 8-bit color values, C0 and C1, may be specified in registers R1 and R2. The low order 4-bit nibble of a color value is drawn in an even bank. The high order color nibble is drawn in an odd bank. When long pels are used, banks 0 and 1 are generally addressed as the frame buffer. When short pels are used, any bank may hold a frame buffer. In this case, the bank parity selects the color nibble used. (See destination pointer section for bank addressing).

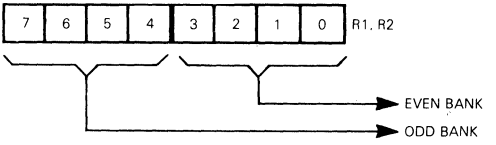


FIGURE 2.4. - COLOR REGISTER

2.4.2. Drawing mode: register R0

The drawing mode defines the transforms to be applied to the pels designated by the drawing commands. There are three drawing modes.

2.4.3. Monochrome mode

Any AREA drawing command, RECTANGLE for instance, defines through its geometric arguments an active set of destination pels, that is to say a set of pels to be modified.

When DMU = 1, this active set is further reduced by the masking mechanism to only these destination pels with a bit mask set.

The active destination pels are then modified according to two elementary transforms coded in R0.

COLOR TRANSFORM:
The color value C of each active pel is modified according to one color transform selected out of four:

- 00 - printed in C0: C ← C0
- 01 - printed in C1: C ← C1
- 10 - printed in "transparent": C ← C
- 11 - complemented: C ← $\overline{C}$.

This yields to a reversible marker mode.

MASK BIT TRANSFORM:
The destination mask bit of each active pel is modified according to one mask transform selected out of four:

- 00 - reset bit mask: M ← 0
- 01 - set bit mask: M ← 1
- 10 - no modification: M ← M
- 11 - complement bit mask: M ← $\overline{M}$.

This scheme allows the color bits and the mask bit of any pel belonging to the active set to be modified independently. The color transform is performed first.

2.4.4. Bichrome mode

A PRINT CHARACTER command is more complex because it involves two different active sets: FOREGROUND and BACKGROUND.

The FOREGROUND is that set of destination pels printed from set elements in the character cell. The BACKGROUND is made of all the remaining pels belonging to the destination window.

When DMU = 1, the FOREGROUND and BACKGROUND are further reduced by the destination masking mechanism. (See figure 2.6).

A bichrome drawing mode is defined by 4 elementary and independent transforms: (see figure 2.5)

- a color transform
- a mask transform

} For the FOREGROUND PELS

- a color transform
- a mask transform

} For the BACKGROUNG PELS

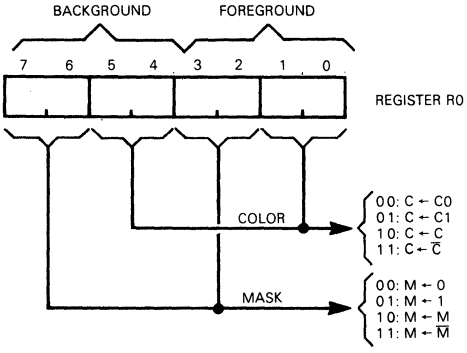
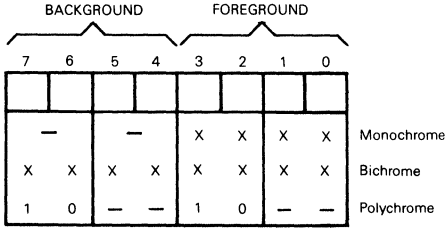


FIGURE 2.5. - DRAWING MODE REGISTER R0



SUMMARY

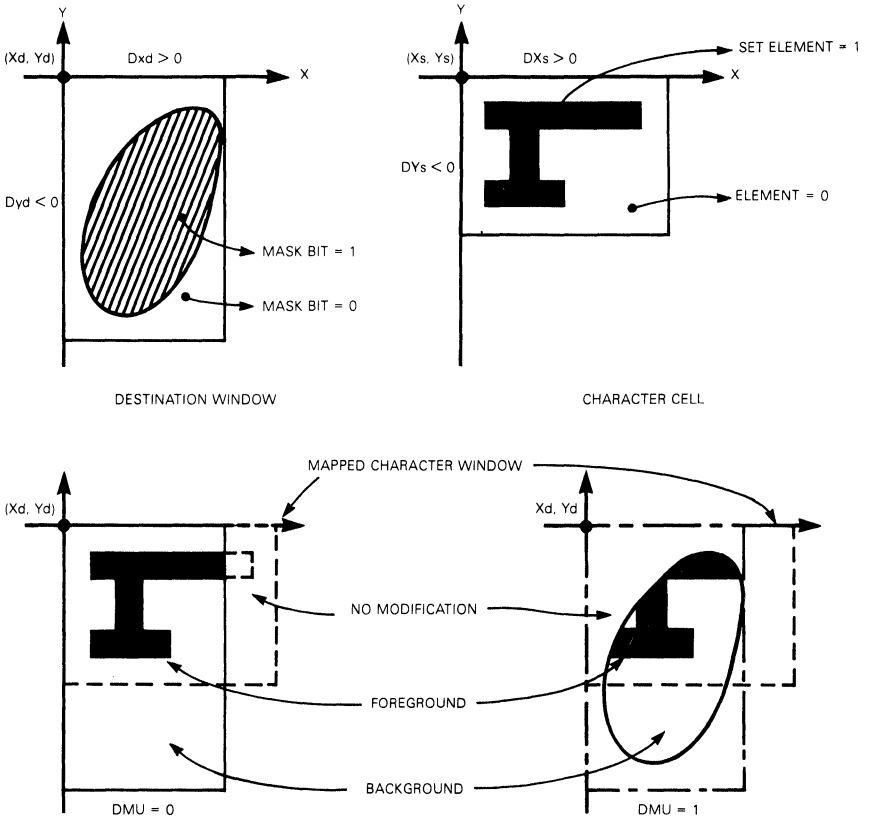


FIGURE 2.6. - PRINT CHARACTER COMMAND

2.4.5. Polychrome mode

A print object command defines a source window through the source pointer:
When SMU = 0, any pel of this window is active, mapped and clipped to the destination window dimension.

When SMU = 1, only pels which have a source mask bit set are active, mapped and clipped to the destination window dimension.

In both cases, when DMU = 1, the active source pels are further reduced by the destination masking mechanism.

Both mask transforms must be programmed at "NO MODIFICATION" for correct operations. (See figure 2.5).

2.4.6. The linear drawing command case

A LINE or ARC drawing command may be executed in any drawing mode depending on the PEN.

When the pen is a DOT, this pel is printed at each active coordinate according to monochrome mode.

When the pen is a CELL, this cell is printed at each active coordinate. In the bichrome mode when the cell is a character, and in the polychrome mode when the cell is an object.

For each active coordinates, the active destination set is defined by the cell dimensions (DXs, DYs).

NOTE: when the cell is an object, SMU is not programmable and is implicitly set. A calculated coordinate is active when the rotated LSB linear texture bit in (R3) is set.

2.4.7. Scaling factor and cell mapping: (See figure 2.7 et 2.8).

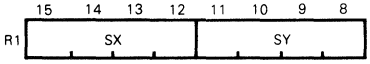
Any cell may be printed with a scaling factor.

This scaling factor is an integer pair Sx, Sy = 1 to 16,

This scaling factor is interpreted with the PRINT CHARACTER, PRINT OBJECT and LINEAR commands when the pen is a cell. The AREA or ACCESS or LINEAR (DOT) commands are never scaled.

The LINEAR (PEN) command should be used with a scaling factor of 1 because the pen is clipped at DXs, DYs.

The scaling factor is first applied to the source cell before mapping and drawing. The drawing and mapping is processed with sign bit of DYd and DYs values. (See figure 2.8).



SX or SY	S
0 0 0 1	1
0 0 1 0	2
⋮	⋮
1 1 1 1	15
0 0 0 0	16

FIGURE 2.7. - SCALING FACTOR

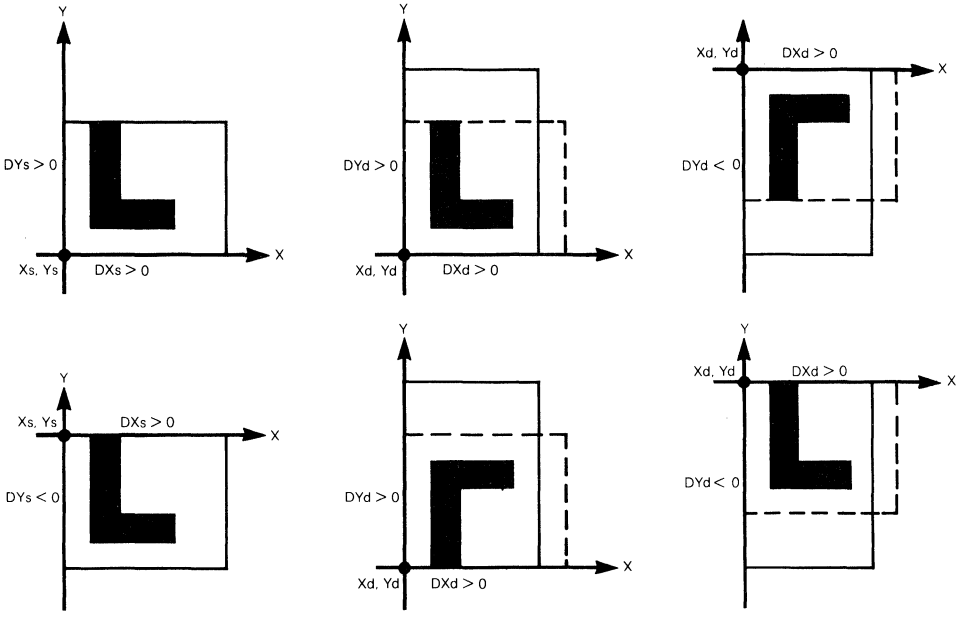
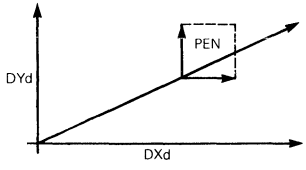


FIGURE 2.8 - CELL MAPPING VERSUS DYd, DYs SIGN

Note:

- DXs is always positive
- The DYs sign mirrors the cell
- DXd must be positive with a PRINT CELL command
- DXd and DYd may get any sign with a LINEAR DRAWING command. If a pen is used, these signs are then irrelevant to the pen drawing. The pen is mapped with positive increment values.



2.5. COMMAND SET OVERWIEW

2.5.1. Linear drawing

LINE (Xd, Yd, DXd, DYd). ARC (Xd, Yd, DXd, DYd, RAD, STOP).

The curve may be drawn with any pen and with any linear texture (register R3). For each set of computed coordinates, R3 is right rotated and the pen is printed when the shifted bit is set.

2.5.2. Area drawing

RECT (Xd, Yd, DXd, DYd)

TRAPEZIUM (Xd, Yd, DXd, DYd, X1)

POLYGON (Xd, Yd, DXd, DYd)

POLYARC (Xd, Yd, DXd, DYd, RAD, STOP)

Either RECT or TRAPEZIUM allows to draw directly all the pels inside the boundary.

Any other closed boundaries may be filled by a 3-step process:
1. The mask bits inside a boundary box must be reset by a RECT command.
2. A sequence of mixed POLYGON and POLYARC commands describing the closed boundary sets the mask bits of the pels inside this boundary.

3. This area may then be painted by a RECTANGLE command defined for a bounding box, with destination masking. It may also be tiled by use of a PRINT CELL command.

Note: the mask bit of any pel lying on the boundary itself is not guaranteed to be set by step 2.

2.5.3. Print cell commands

PRINT CELL (Xd, Yd, DXd, DYd; Xs, Ys, DXs, DYs)

The cell addressed by Xs, Ys, DXs, DYs is scaled then printed at location Xd, Yd and clipped at the dXd, dYd dimensions.

When dXd, dYd is much larger than DXs, DYs the command may be parametered for repeat drawing.

These commands may also be parametered for destination mask use.

Further more the PRINT OBJECT command may be parametered for source mask use.

These commands have a wide range of applications: text drawing, area tiling, print or move objects, scale and move viewports.

Note: an underlined cell is drawn when the MSB of R23 is set.

2.5.4. Access commands

- LOAD VIEWPORT (Xs, Ys, DXs, DYs).
- SAVE VIEWPORT (Xs, Ys, DXs, DYs).
- MODIFY VIEWPORT (Xs, Ys, DXs, DYs).

These commands provide sequential access to a viewport in a frame buffer from the microprocessor data base.

Data are transferred to/from the display memory, word sequentially.

The R14 to R17 registers are used as a two memory word FIFO (memory word is 8 short pels, i.e. 4 bytes).

The source pointer (R20 - R23) is used to address the viewport for all access commands.

When long pels are used, the command must be executed once more when the bank number in R20 has been updated.

2.5.5. Command execution

Each on-chip 16-bit register has four addresses. One address is used for plain read or write. The other addresses are used to initiate command execution automatically on completion of the register access.

This scheme allows the command code and its arguments to be loaded or modified in any order. An incremental line drawing command, for example, may be executed again and again with successive incremental dimensions and whithout need to reload the command code itself.

As soon as a command execution is started, the FREE bit is cleared in the STATUS register. This bit is automatically set when the execution is completed.

The commands are generally executed only during retrace intervals. However full time execution is possible when either the display is disabled or video RAM components are used.

2.5.6. Status register (See figure 2.9).

This register holds four read-only status bits:
- FREE: this status bit is set when no execution is pending
- VS: vertical synchronisation state
- SEM: this status bit is set when the FIFO memory word is inaccessible to the microprocessor during a viewport transfer
- NSEM: this status bit is set when the FIFO memory word is accessible to the microprocessor during a viewport transfer.

Each of these status bits is maskable. The masked status bits are NORed to the $\overline{IRQ}$ output pin.

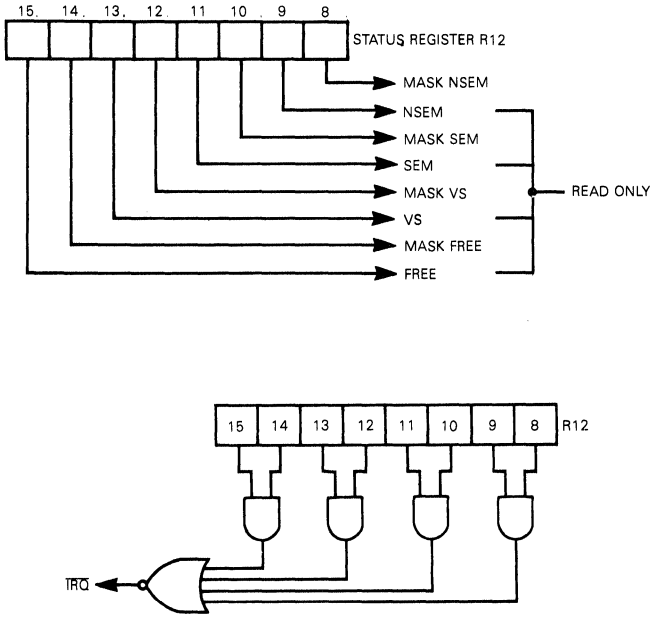


FIGURE 2.9. - STATUS REGISTER

SECTION 3
MICROPROCESSOR INTERFACE

3.1. INTRODUCTION

The TS68483 is directly compatible with any popular 8 or 16-bit host microprocessor; either Motorola type (6809, 68008, 68000) or Intel type (8088, 8086).

The host microprocessor has direct access to any of the twenty four 16-bit on-chip registers through the microprocessor interface pins:

- D(0:15): 16 bidirectional data pins.
- A(0:7): 8 address inputs.
- $\overline{AE}$, $\overline{DS}$, R/$\overline{W}$, $\overline{CS}$: 4 control inputs.

The twenty four registers are mapped in the host addressing space as 256 byte addresses. (See figure 3.2)

- A(1:5) select one out of 24 registers.
- AO selects the low order byte (A0 = 1) or the high order byte (A0 = 0) of the selected register.
- A(6:7) provide the command execution condition.

The host microprocessor bus may be either 8 or 16-bits wide and may be address/data multiplexed or not.

The two flags MB and BW in the CONFIGURATION register R10 allow the data bus size and multiplexed/non-multiplexed organization to be specified. (See figure 3.1).

TYPE OF MPU BUS		CONF. REG.		TS68483 PINS					
		BW	MB	AE	$\overline{DS}$	R/$\overline{W}$	AO	A(1:7)	D(8:15)
NON MUX	16-bit (68000)	0	0	1	$\overline{UDS}$ or $\overline{LDS}$	R/$\overline{W}$	O	A(1:7)	D(8:15)
	8-bit (68008)	1	0	1	$\overline{DS}$	R/$\overline{W}$	AO	A(1:7)	D(0:7)
MUX	16-bit (8086)	0	1	ALE	$\overline{RD}$	$\overline{WR}$	O	AD(1:7)	AD(8:15)
	8-bit (8088)	1	1	ALE	$\overline{RD}$	$\overline{WR}$	$\overline{ADO}$	AD(1:7)	AD(0:7)

FIGURE 3.1. - MPU SELECTION

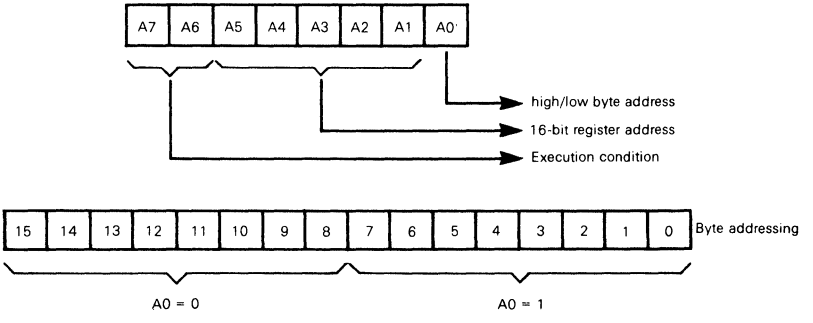


FIGURE 3.2. - ON-CHIP ADDRESS AND BYTE PACKING

3.2. HARDWARE RECOMMENDATIONS (See timing diagrams 1 and 2).

A0-PIN:

1. When using a 16-bit data bus, the A0 input pin must be grounded. No single byte access can be performed.

2. In order to conform with the high byte/low byte on-chip packing, the A0 input pin must be inverted when using an 8-bit bus Intel type microprocessor (8088 for example).

A(1:7), D(0:7), D(8:15) pins:

1. With any 8-bit data bus, the D (0:7) and D (8:15) pins must be paired in order to demultiplex the low order data bytes and the high order data bytes.

2. When using address/data multiplexed bus, the D (0:7) pins are paired with A (0:7) in order to demultiplex data from address.

$\overline{AE}$, $\overline{DS}$, R/$\overline{W}$, $\overline{CS}$:

See pin description.

3.3. SOFTWARE RECOMMENDATIONS

1. The CONFIGURATION register R10 must be first initialized.

The BW 15 flag is interpreted by the bus interface to recognize an 8-bit/16-bit data bus.

The MB and BW 15 flags are used to decide when to initiate a command execution.

2. Each register byte has 4 addresses in the microprocessor memory map. These 4 addresses differ only by A(6:7). This scheme allows a 68008 programmer to read or write any data type (byte, word, long word) and automatically initiate or not a command execution at the end of this transfer. The transfer lasts one, two or four bus cycles.

A 68000 programmer is restricted to only word and long word data types. (See figure 3.3).

ADDRESS		EXECUTION CONDITION	DATA TYPE TRANSFER	
A7	A6		8-bit data bus	16-bit data bus
0	0	No Exec	Any type	Any type
0	1	Exec after a bus cycle	1 byte	1 word
1	0	Exec after 2 bus cycles	1 word	1 long word
1	1	Exec after 4 bus cycles	1 long word*	ILLEGAL

NOTE: Word transfer must respect word boundary.
Long word transfer must respect long word boundary.
*Not available with 8088 MPU type.

FIGURE 3.3. - COMMAND EXECUTION CONDITION

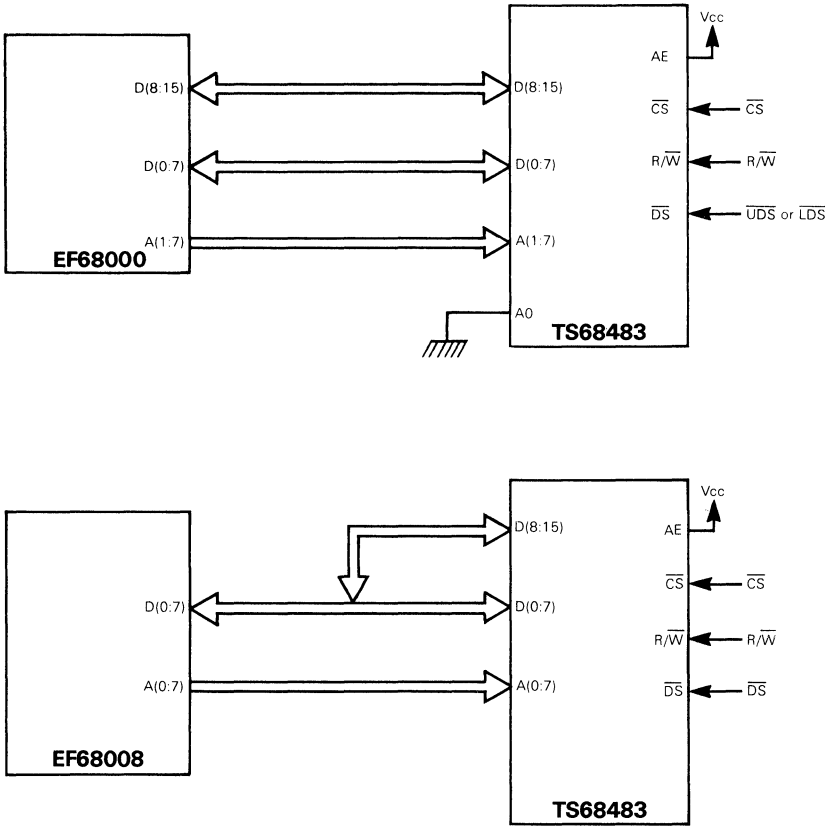


FIGURE 3.4. - INTERFACE WITH EF68000/68008 MPU

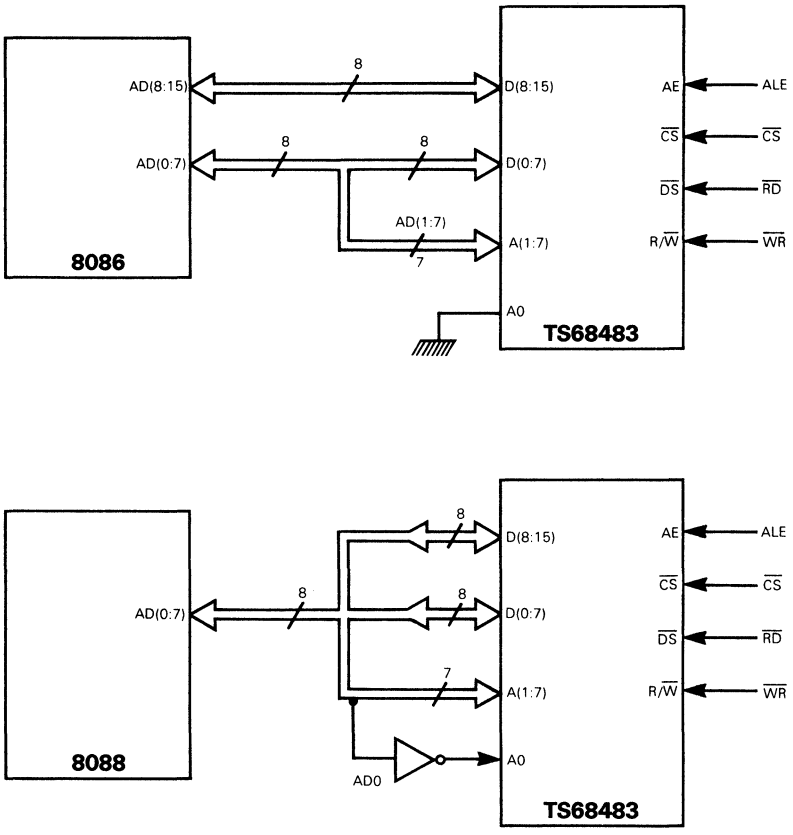


FIGURE 3.5. - INTERFACE WITH 8086/8088 MPU

SECTION 4
THE VIDEO TIMING GENERATOR
RAM REFRESH AND DISPLAY PROCESS

4.1. INTRODUCTION

The Video Timing Generator is completely synchronous with the CLK input, which provides a pixel shift frequency (12 to 15 MHz). The Video Timing Generator:

- delivers the blanking signal (BLK), the horizontal (HS) and vertical (VS) synchronization signals on respective ouput pins,
- schedules the memory time allocated to the display process, dynamic RAM refresh and command execution,
- is fully programmable
- can be synchronized with an external composite video sync signal connected to the SYNC IN input.

4.2. SCAN PARAMETERS (See table 1 and timing diagram 5)

4.2.1. Timing units

The time unit of any vertical parameter is the scan line.

The time unit of any horizontal parameter is the memory cycle, which is 8 periods of the CLK input signal.

These two parameters are internally programmed:

- Horizontal sync pulse duration = 7 cycles
- Vertical sync pulse duration = 2.5 lines.

4.2.2. Blanking interval

The blanking interval starts:

- at the leading edge of the vertical sync pulse. Vertical blanking interval actual duration is 2.5 lines more than the programmed value.
- two cycles before the leading edge of the horizontal sync pulse. The actual horizontal blanking interval duration is 3 cycles more than the programmed value.

NOTE: During the programmed blanking interval, the video output pins P(0:3) are forced low.

4.2.3. Porch and margin color

During the porch interval, the programmable margin color is displayed on the P(0:3) outputs.

The display process may be disabled by setting DPD flag. This will be interpreted as a porch extension.

4.2.4. Memory time sharing (See figure 4.1)

The Video Timing Generator allocates memory cycles to either the display process, RAM refresh or command execution. In this respect, the scan lines per field are split between:

- the DWY displayable lines.

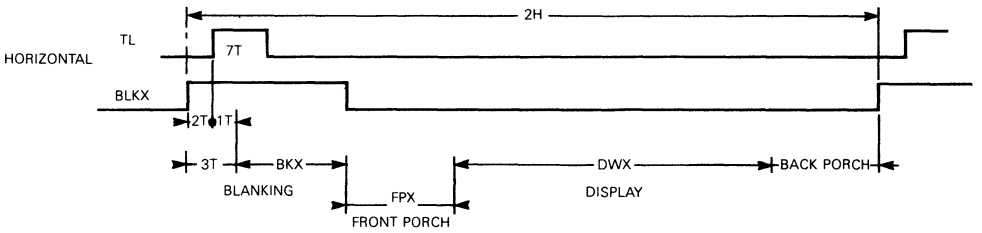
When VRE = 0, Video RAMs are not used.

The DWY x DWX cycles in the display interval are allocated to the display process when it is enabled (DPD = 0). When the display process is disabled, these cycles are allocated as for non displayable lines.

When VRE = 1, one cycle per display line is allocated to the display process. Other cycles are allocated as for non displayable lines. The last period of the BLKX signal may be used to load the internal video RAM shift register.

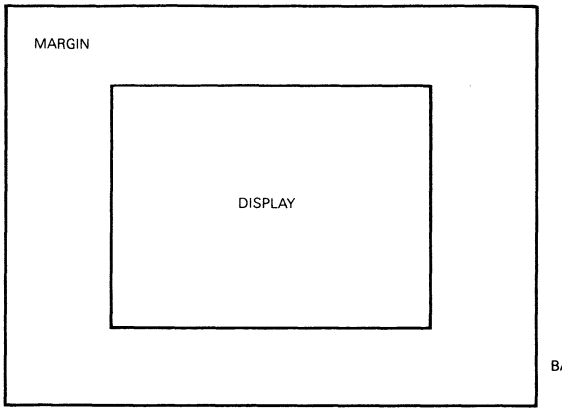
- the non displayable lines. In one out of nine non displayable lines, DWX cycles are allocated to the refresh process when it is enabled (RFD = 0).

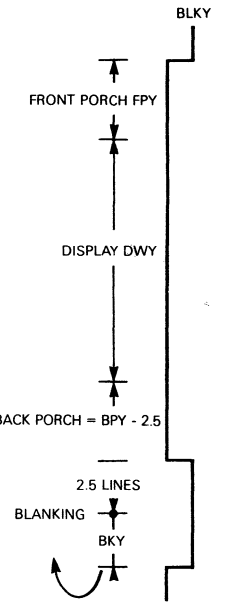
- In Float cycle, an external X address must be provided. The Y address is still provided on ADM(0:7) and Y(0:2), while ADM)8:15) are in high impedance state.



HORIZONTAL MINIMUM:
NUMBER OF CYCLES

BKX	4
FPX	3
DWX	3
H	19





VERTICAL MINIMUM:
NUMBER OF LINES

FPY	1
DWY	1
BPY	3
BKY	1

FIGURE 4.1. - VIDEO PROGRAMMING

4.2.5. Command access ratio

This allocation scheme leaves about 50% of the memory bandwidth for command access when programming a standard TV scan. This ratio drops to the 30% range when a better monitor is in use (32 µs out of 43 µs displayable per line, 360 lines out of 390 for a 60 Hz field rate). The higher resolution means more memory accesses in order to edit a given percentage of the screen area. In this case Video RAMs are very helpful to keep 90% of the memory bandwidth available for command access.

4.3. DISPLAY PROCESS

The Video Timing Generator allocates memory cycles to the Display Processor in order to read the Display Viewport from memory. The Display Viewport upper left corner address is programmable through DIB, YOR and XOR. The display viewport dimensions are related to the display interval of DWY lines by DWX cycles per field.

4.3.1. Y addresses

When INE = 0, the fields are not interlaced. The Y Display Viewport address is initialized with YOR at the first displayable line then decremented by 1 at each scan line. The Display Viewport is thus DWY pel high.
When INE = 1, the fields are interlaced. The Y Display Viewport address is initialized as shown in the table below. It is then decremented by two at each scan line. The viewport is thus 2 x DWY pel high

	EVEN FIELD	ODD FIELD
YOR EVEN	YOR	YOR + 1
YOR ODD	YOR – 1	YOR

Y Display Viewport address initialization when INE=1.

4.3.2. X addresses and MODX flags

The X Display Viewport address is initialized with XOR at the first displayable cycle of each displayable line. It is then incremented at each subsequent cycle according to MODX flags.

MODX1	MODX0	X INCR	VIDEO SHIFT REGISTER	MEMORY CYCLE TYPE
0	0	+1	INTERNAL	READ
0	1	+1	EXTERNAL	DUMMY READ
1	0	+2	EXTERNAL	DUMMY READ
1	1	–	EXTERNAL	FLOAT

– In internal mode, the Display Viewport is 8. DWX pel wide. The on-chip video shift register are used.

– In Dummy read, the memory is read but the on-chip video shift registers are not loaded, instead they retain their margin color. External video shift registers are presumed to be loaded by either 8 pels or 16 pels per cycle according to the programmed increment value.

– In Float cycle, an extenal X address must be provided. The Y address is still provided on ADM(0:7) and Y(0:2), while ADM(8:15) are in high impedance state.

NOTE: See Memory Organization and Memory Timing for further details on the memory cycles.

4.3.3. The Video RAM case (VRE = 1)

In this case, the last cycle of the horizontal blanking interval is systematically allocated to the display process for DWY scan lines per field.

This cycle bears the scan line address, the bank number and the X address which is always XOR.

MODX must be programmed to use external shift register (Dummy read).

4.3.4 PAN and TILT

The host can tilt or pan the Display Viewport through the frame buffer by modifying YOR or XOR arguments. Panning is performed on 8 pel boundaries.

4.4. DYNAMIC RAM REFRESH

No memory cycles are explicity allocated to the RAM refresh when RFD = 1.

When VRE = 0 and DPD = 0, the Display Process is supposed to be able to over-refresh dynamic components. This can be done by careful logical to component address mapping. During the remaining non displayable lines, the Display Viewport address continues to be incremented: Y address on each line according to INE, X address initialized by XOR then incremented according to MODX. This Display viewport address is allowed to address the memory for DWX cycles in only one line out of nine for refresh purposes.

When VRE = 1 or DPD = 1, any line is processed as a non displayable line with respect to the refresh process.

4.5. CONFIGURATION AND EXTERNAL SYNCHRONIZATION

The R10 register holds eight configuration flags. Six of these flags are dedicated to the Video Timing Generator.

- SSP: this flag selects the synchronization output pin configuration:

- NPC, NHVS, NBLK: these three flags invert the PC/HS, HVS/VS and BLK outputs respectively. (Ex.: When NBLK = 1 blanking is active high).

The SYNC IN input pin provides an external composite synchronization signal input from which a Vertical Sync In (VSI) signal is extracted. The SYNC IN signal is sampled on-chip at CLK frequency. Its rising sampled edge is compared to the leading edge of HS. A PC comparison signal is externally available (see SSP and NPC flags).

VSIE: this flag enables VSI to reset the internal line count.

HSIE: this flag enables the rising edge of SYNC IN to act directly on the Video Timing Generator. When the leading edge of HS does not match at 1 clock period a rising edge of SYNC IN, one extended cycle is performed (nine clock periods instead of eight).

FLAG	OUTPUT PINS	
	PC/HS	HVS/VS
SSP = 1	HS	VS
SSP = 0	PC	HVS

NAME	NUMBER OF BITS	MINIMUM VALUES	REGISTER	DESCRIPTION	FUNCTION
DWY	10	1	R9	Number of display lines per field	Vertical scan
INE	1	–	R8	Interlace Enable when INE = 1	
BKY	5	1	R8	Number of lines in vertical blanking - 2,5	
FPY	5	1	R7	Number of lines in vertical front porch	
BPY	8	3	R6	Number of lines in vertical back porch + 2.5	
H	6	19	R6	Number of double cycles per line	Horizontal scan
FPX	4	3	R8	Number of cycles in horizontal front porch	
BKX	4	4	R8	Number of cycles in horizontal blanking - 3	
DWX	7	3	R7	Number of cycles of the display window	
XOR	8	–	R4	X, Y and bank logical address in the display memory of the display viewport upper left corner.	Display process
YOR	11	–	R5		
DIB	2	–	R4		
MODX	2	–	R9	Selection of the X addressing mode	
MC	4	–	R4	Margin color	
RFD	1	–	R7	RAM refresh disable when RFD = 1	Memory time sharing
DPD	1	–	R7	Display process disable when DPD = 1	
VRE	1	–	R8	Video RAM enable when VRE = 1	

NOTE: One cycle = 8 periods of CLK clock.

TABLE 1.

SECTION 5

MEMORY ORGANIZATION

5.1. INTRODUCTION

The display memory is logically organized as four banks of 4-bit planes. Thus a bit address in the display memory is given by the quadruplet:

- B = bank number, from 0 to 3
- Z = plane number, from 0 to 3
- X = bit address into the plane, from 0 to 2047
- Y = bit address into the plane, from 0 to 2047.

In one memory cycle (8 CLK periods), the controller can access a memory word. This 32-bit memory word holds one byte from each plane in a given bank. In order to address this memory word, the controller supplies:

- B(0:1): binary value of the bank number
- X(3:10): binary value of the word address
- Y(0:10): binary value of the word address.

Z and X(0:2) are not supplied. They give only a bit address in a memory word.

5.2. MEMORY CYCLES

24 pins are dedicated to the memory interface.

- ADM(0:15): these 16 bidirectional pins are multiplexed three times during a memory cycle (see Timing Diagram 3):
  - TA: address period. Output of the X(3:11) and Y(3:11) address
  - T0: even data period. The even Z bytes are either input or output
  - T1: odd data period. The odd Z bytes are either input or output.

- Y(0:2): three LSB Y address output pins (non-multiplexed)

- B(0:1): two bank address output pins (non-multiplexed)

- CYS: Cycle start strobe output (non-multiplexed).

CYS is at CLK/8 frequency. A CYS pulse is delivered only when a command, display or refresh cycle is performed.

- CYF(0:1): Two cycle status outputs (non-multiplexed). Four cycle types are defined:
  - Command read
  - Command write
  - RAM refresh
  - Display access.

Because several options may be selected for RAM refresh and display access by the MODX and VRE flags (see Video Timing Section), there are more than four memory cycle types (see Timing Diagram 3 and Table 2).

5.3. DISPLAY MEMORY DESING OVERVIEW

The display memory implementation is application dependant. The basic parameters are:

- the number of pixels to be displayed Nx.Ny
- the number of bits per pel
- the vertical scanning frequency, which must be picked in the 40 Hz to 80 Hz range (non interlaced) or in the 60 Hz to 80 Hz range (interlaced).

This yields a rough estimate of the pixel frequency. When the pixel frequency is in the 12 to 15 MHz range and 4 bits per pixel or least are required, the on-chip video registers and standard dynamic RAM components may be used. When higher pixel rates or up to 8 bits per pixel are required, the designer must provide external shift registers. Video RAM components may also be considered.

In either case, the user must design:

- A memory block. This is the hardware memory building block. It includes the video shift registers if on-chip VSR cannot be used. It implies a RAM component choice.

- An Address Mapper, which maps the logical address into hardware address: block selection, Row Address (RAD), Column Address (CAD).
- A memory cycle controller. This controller monitors the CYF and CYS output pins from TS68483 and block address from the Mapper. It provides:

· The CLK signal to the TS68483 and a shift clock SCLK when external video shift registers are used
· $\overline{RAS}$, $\overline{CAS}$, $\overline{OE}$, R/$\overline{W}$ signals to the memory blocks
· RAD and CAD Enable signals to the Mapper.

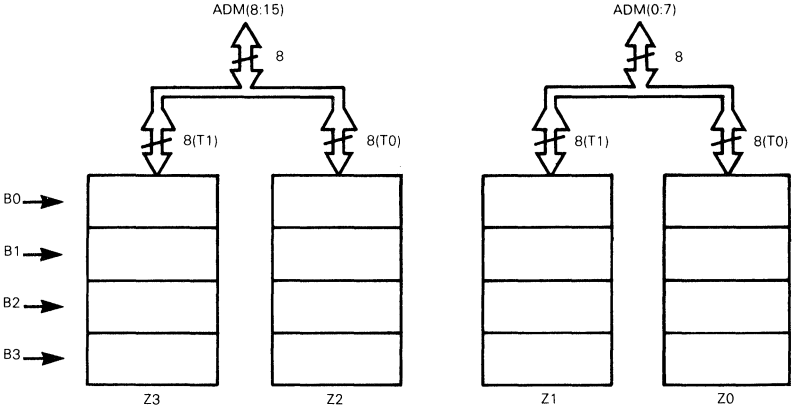
5.3.1. Frame buffer (See figure 5.1.)

A byte wide organization of each bit plane is required. Obviously a bit plane must contain the Display Viewport size. A straight organization implements only one bit plane per block.

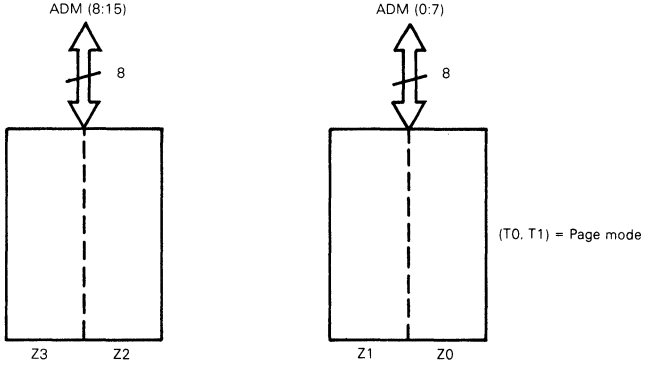
It may be cost effective to implement several bit planes per block. Two basic schemes may be used:

- One block, one Z: several bit planes, belonging to different banks, but addressed by the same Z, share a given block. There is little time constraint if any.

- One block, two Z: two bit planes, belonging to the same bank share a given block. In this case, this block must be accessed twice during a memory cycle. This can be solved by two successive page mode accesses.



ONE BLOCK-ONE Z



ONE BLOCK-TWO Z

TYPICAL BLOCK SIZE	16K x 8	32K x 8	64K x 8	256K x 8
ONE BLOCK-ONE BIT PLANE	512 x 256	512 x 512	1024 x 512	2048 x 1024
ONE BLOCK-TWO BIT PLANES	256 x 256	512 x 256	512 x 512	–

COMPONENTS:
64K BITS: 16K x 4 or 64K x 1
256K BITS: 32K x 8, 64K x 4, 256K x 1
VIDEO RAM: 64K x 1, 64K x 4

FIGURE 5.1. - FRAME BUFFER ORGANIZATION

OUTPUT PINS		FUNCTION	MODX FLAGS 1 0	MULTIPLEXED ADM			CYCLE TYPE
CYF1	CYF0			TA	T0	T1	
1	0	COMMAND READ	–	Y, X	Z0, Z2	Z1, Z3	READ
1	1	COMMAND WRITE	–	Y, X	Z0, Z2	Z1, Z3	WRITE
0	1	DISPLAY	0 0 0 1	Y, X Y, X	Z0, Z2 –	Z1, Z3 –	READ DUMMY READ + 1
0	0	REFRESH	1 0 1 1	Y, X Y, Hi-Z	– –	– –	DUMMY READ + 2 FLOAT X

Refresh: dummy read cycle is performed.

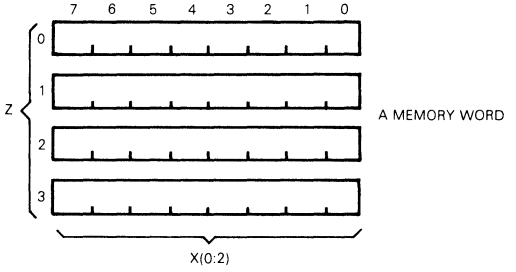


TABLE 2. - MEMORY CYCLE TYPES

ADMS MULTIPLEXED PINS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TA: ADDRESS PERIOD	10			X				3	10			Y				3
T0: EVEN Z BYTE PERIOD	7			Z = 2				0	7			Z = 0				0
T1: ODD Z BYTE PERIOD	7			Z = 3				0	7			Z = 1				0
	HIGHER BYTES								LOWER BYTES							

FIGURE 5.2. - THE MULTIPLEXING SCHEME

5.3.2. Masking planes

Masking planes are very useful for general purpose area filling or clipping. It may be practical to use one or two planes smaller than the color bit plane if they cyclically cover a frame buffer.

The masking planes must be in bank 3.

5.3.3. Objects and characters

Objects may be located in unused parts of the frame buffer.

Character generators can be implemented in any plane of any bank. They can also be implemented in ROM. In this case, plane Z = 1 or 3 offer relaxed access time requirements.

5.4. EXAMPLES

Figure 5.3. gives the schematic for a 512 x 384 non interlaced application. A CLK signal in the 13 to 15 MHz range should produce a 50 to 60 Hz refresh rate. The on-chip video shift registers may be used if no more than four bits per pixel are required. One 64K x 8 memory block may be implemented using either eight 64K x 1 or two 64K x 4 components. One memory block holds two 512 x 384 color bit planes.

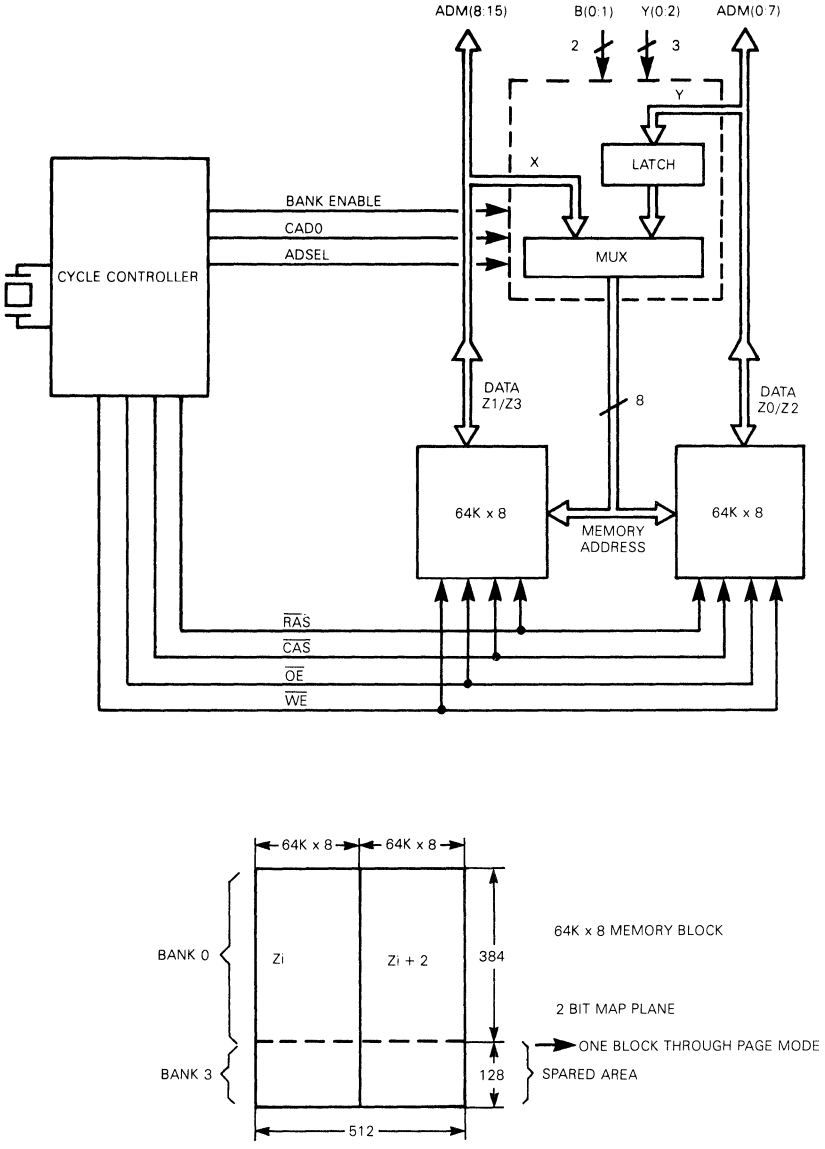


FIGURE 5.3. - MEMORY ORGANIZATION FOR 512 x 384 APPLICATION

SECTION 6
TIMING DIAGRAM

6.1. MICROPROCESSOR INTERFACE

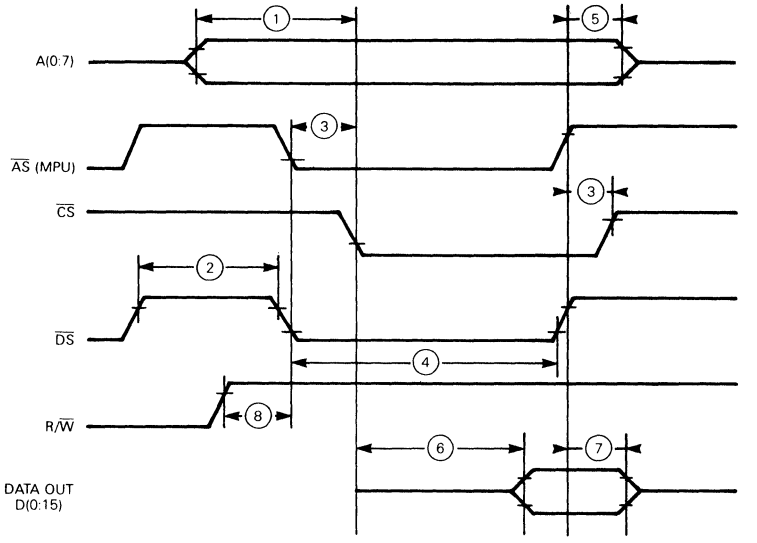
TS68483 has an eight bit address bus and a sixteen bit data bus. Little external logic is needed to adapt bus control signals from most of the common multiplexed or non-multiplexed bus microprocessors.

Microprocessor interface timing: A(0:7), D(0:15), AE, $\overline{DS}$, $\overline{CS}$, R/$\overline{W}$
V_{CC} = 5.0 V ± 5%, T_A = T_L to T_H, C_L = 100 pF on D(0:15)
Reference levels: V_{IL} = 0.8 V and V_{IH} = 2 V on all inputs
V_{OL} = 0.4 V and V_{OH} = 2.4 V on all outputs

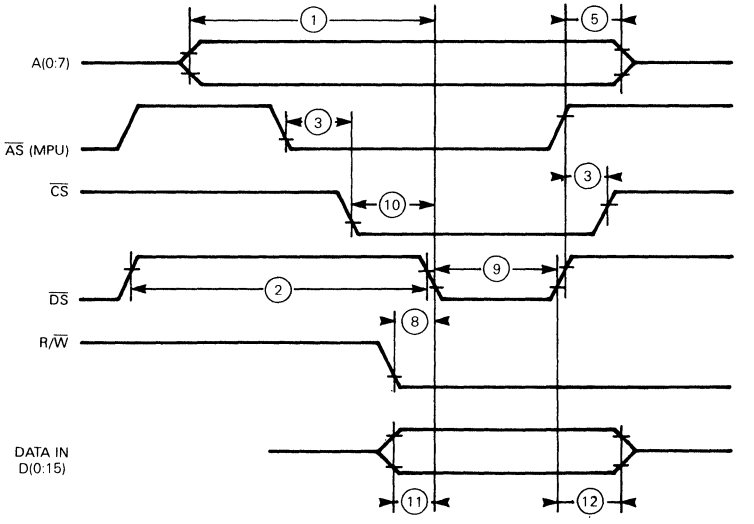
UNMUX MODE

Id. numb.	Characteristic	Min.	Max.	Unit
1	Address set up time from $\overline{CS}$	30	–	ns
2	Data strobe width (High)	150	–	ns
3	$\overline{AS}$ set up time from $\overline{CS}$	0	–	ns
4	Data strobe width-low (Read cycle)	240	–	ns
5	Address hold time from $\overline{DS}$	0	–	ns
6	Data access time from $\overline{CS}$ (Read cycle)	–	210	ns
7	$\overline{DS}$ inactive to high impedance state (Read cycle)	10	100	ns
8	R/$\overline{W}$ set up time from $\overline{DS}$	20	–	ns
9	$\overline{DS}$ width-low (Write cycle)	110	–	ns
10	$\overline{CS}$ set up time from $\overline{DS}$ active (Write cycle)	0	–	ns
11	Data in set up time from $\overline{DS}$ active (Write cycle)	10	–	ns
12	Data in hold time from $\overline{DS}$ inactive (Write cycle)	30	–	ns

UNMUX MODE



READ CYCLE



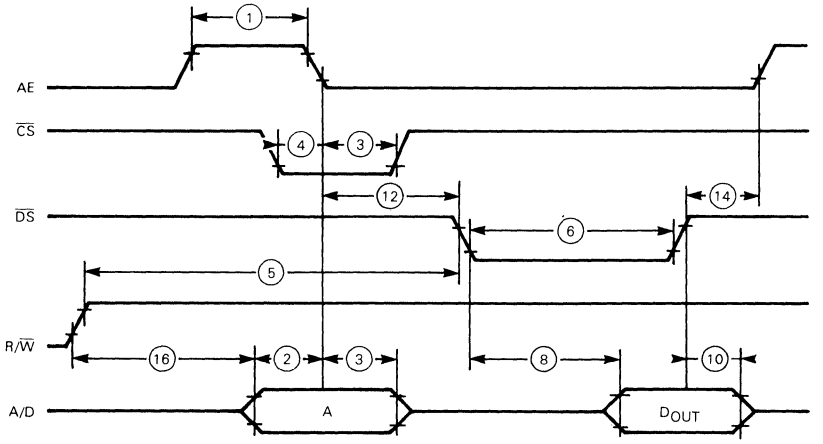
WRITE CYCLE

Microprocessor interface timing: A(0:7), D(0:15), AE, $\overline{DS}$, $\overline{CS}$, R/$\overline{W}$
V_{CC} = 5.0 V ± 5%, T_A = T_L to T_H, C_L = 100 pF on D(0:15)
Reference levels: V_{IL} = 0.8 V and V_{IH} = 2 V on all inputs
V_{OL} = 0.4 V and V_{OH} = 2.4 V on all outputs

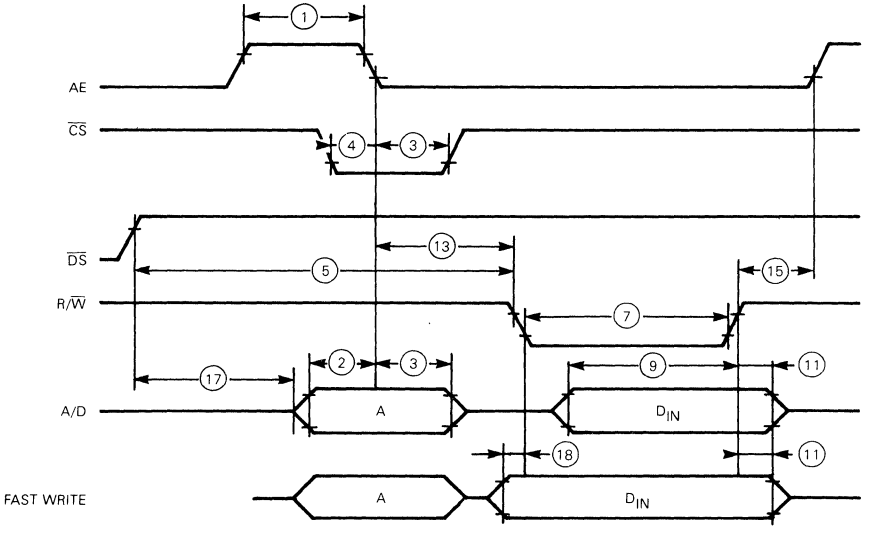
MUX MODE

Id. numb.	Characteristic	Min.	Max.	Unit
1	AE width high	90	–	ns
2	Address set up time to AE inactive	55	–	ns
3	Address and $\overline{CS}$ hold time to AE inactive	55	–	ns
4	$\overline{CS}$ set up time to AE inactive	40	–	ns
5	$\overline{DS}$ and R/$\overline{W}$ high	150	–	ns
6	$\overline{DS}$ width-low (Read)	240	–	ns
7	R/$\overline{W}$ width-low (Write)	110	–	ns
8	Data access time from $\overline{DS}$ (Read)	–	210	ns
9	Data in set up time from R/$\overline{W}$ inactive (Write)	150	–	ns
10	$\overline{DS}$ inactive to high impedance state (Read)	10	100	ns
11	Data in hold time from R/$\overline{W}$ inactive (Write)	30	–	ns
12	AE inactive to $\overline{DS}$ active	20	–	ns
13	AE inactive to R/$\overline{W}$ active	20	–	ns
14	$\overline{DS}$ inactive to AE active	10	–	ns
15	R/$\overline{W}$ inactive to AE active	10	–	ns
16	R/$\overline{W}$ inactive to next address valid	100	–	ns
17	$\overline{DS}$ inactive to next address active	100	–	ns
18	Data in set up time from R/$\overline{W}$ active (Fast write cycle)	10	–	ns

MUX MODE



READ CYCLE



WRITE CYCLE

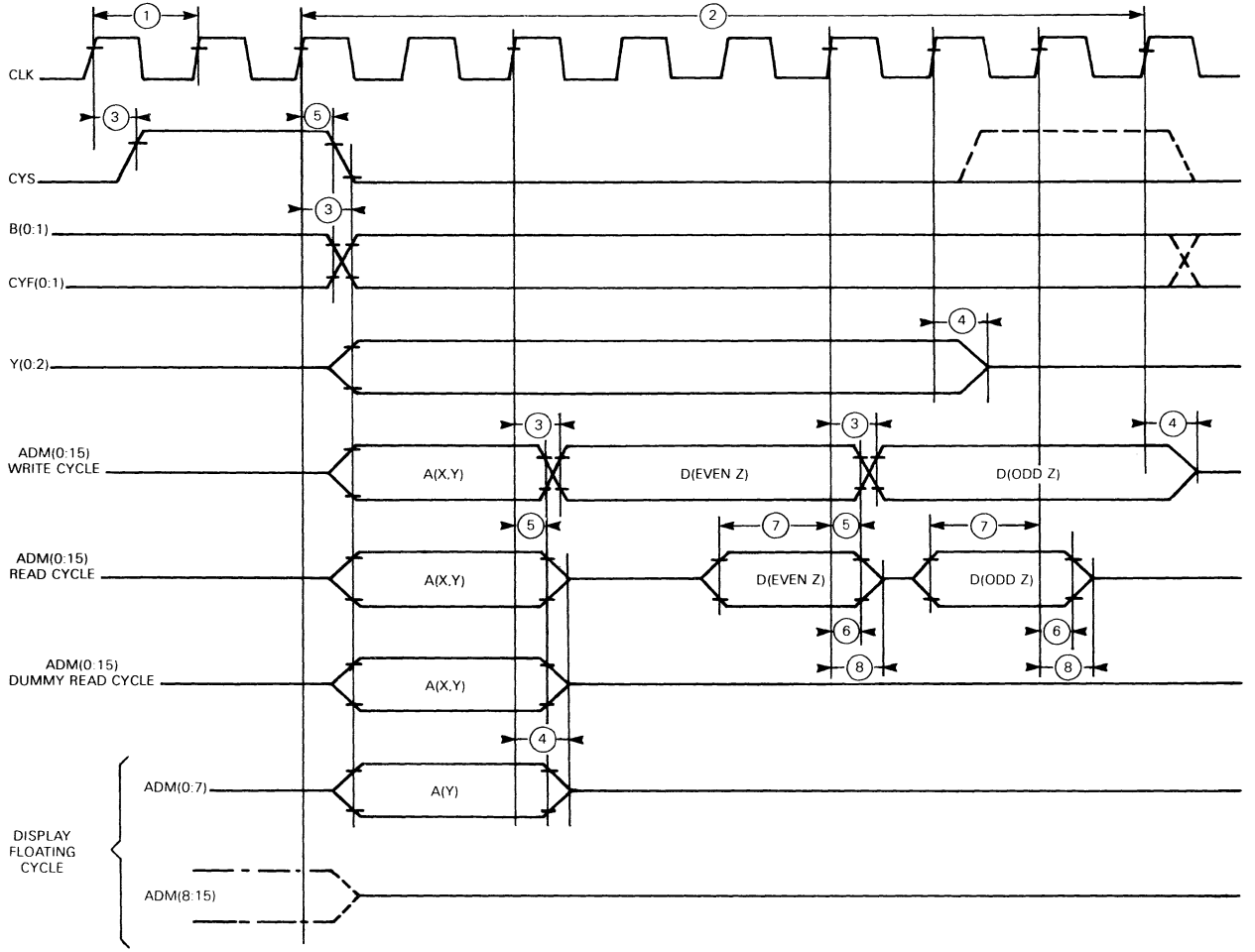
6.2. MEMORY INTERFACE

ADM(0:15), B(0:1), CYF(0:1), Y(0:2), CYS
V_{CC} = 5.0 V ± 5%, T_A = T_L to T_H
CLK duty cycle = 50%, period T
Reference levels: V_{IL} = 0.8 V and V_{IH} = 2 V, V_{OL} = 0.4 V and V_{OH} = 2.4 V

IDENT NUMBER	CHARACTERISTIC	MIN	MAX	UNIT
1	TCLK clock period	66	166	ns
2	Memory cycle time (T = 8 X TCLK)	–	–	ns
3	Output delay time from CLK	–	40	ns
4	Output data HI-Z time from CLK	–	40	ns
5	Output hold time from CLK	10	–	ns
6	Input data hold time from CLK (read cycle)	20	–	ns
7	Input data set up time from CLK (read cycle)	30	–	ns
8	Input data HI-Z time from CLK	–	TCLK	ns

NOTE: All timing is referenced to the rising edge of CLK. (See timing diagram 3)

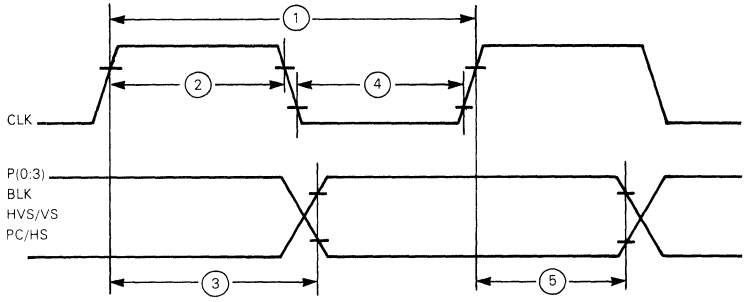
MEMORY INTERFACE



6.3. VIDEO INTERFACE

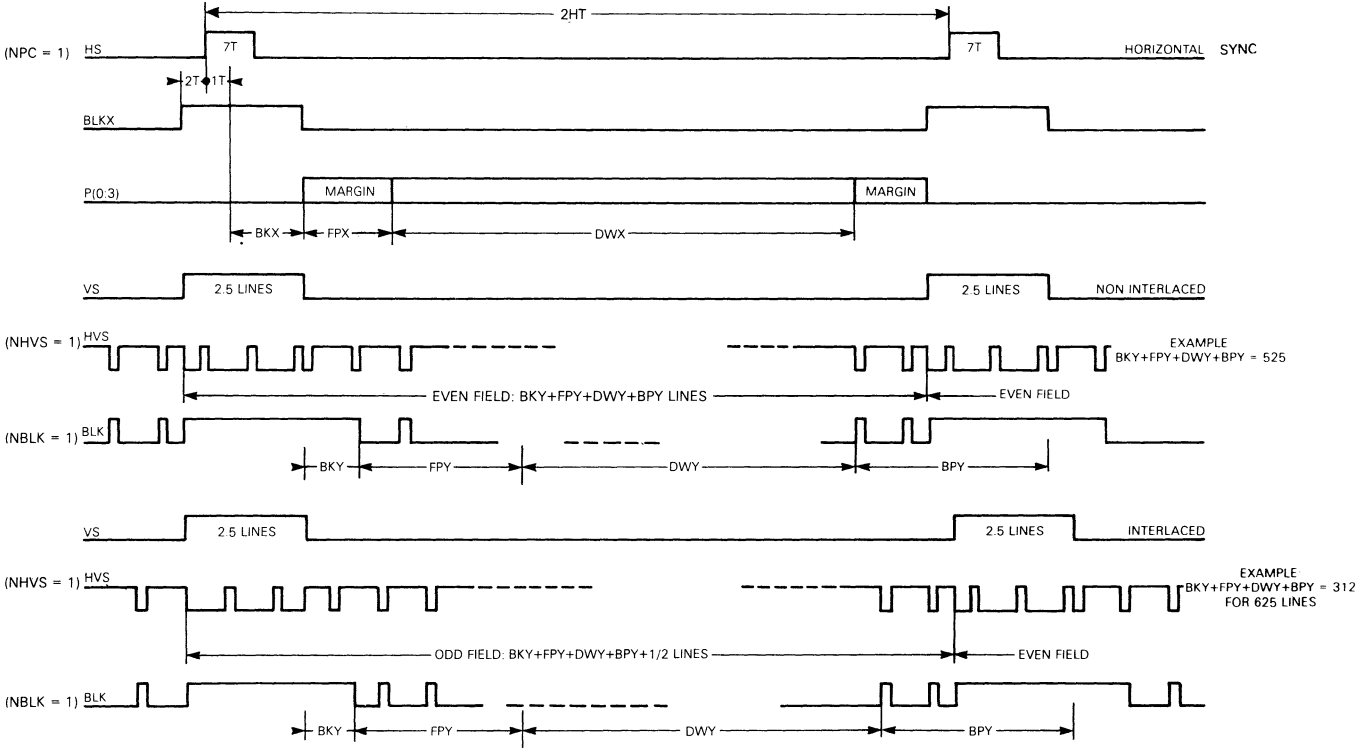
P0, P1, P2, P3, BLK, HVS/VS, PC/HS
V_{CC} = 5.0 V ± 5%, T_A = T_L to T_H, CLK duty cycle = 50%
Reference levels: $V_{I.L}$ = 0.8 V and V_{IH} = 2 V, V_{OL} = 0.4 V and V_{OH} = 2.4 V, C_L = 50 pF

TIMING DIAGRAM 4.



IDENT NUMBER	CHARACTERISTIC	MIN	TYP	MAX	UNIT
1	TCLK : CLK period	66	83	166	ns
2	CLK high pulse width	28	–	–	ns
3	Output delay from CLK rising edge	–	–	40	ns
4	CLK low pulse width	28	–	–	ns
5	Output hold time	10	–	–	ns

SYNCHRONIZATION SIGNAL OUTPUTS



MAXIMUM RATINGS

Stresses above those hereby listed may cause permanent damage to the device. The ratings are stress ones only and functional operation of the device at these or any conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. Standard MOS circuits handling procedure should be used to avoid possible damage to the device.

RATING	SYMBOL	VALUE	UNIT
Supply voltage	V_{CC}*	-0.3 to 7.0	V
Input voltage	V_{in}*	-0.3 to 7.0	V
Operating temperature range TS68483C TS68483V TS68483M	T_A	T_L to T_H 0 to 20 - 40 to + 85 - 55 to + 125	°C
Storage temperature range	T_{stg}	-55 to +150	°C
Max power dissipation	P_{Dm}	1.5	W

* With respect to V_{SS}

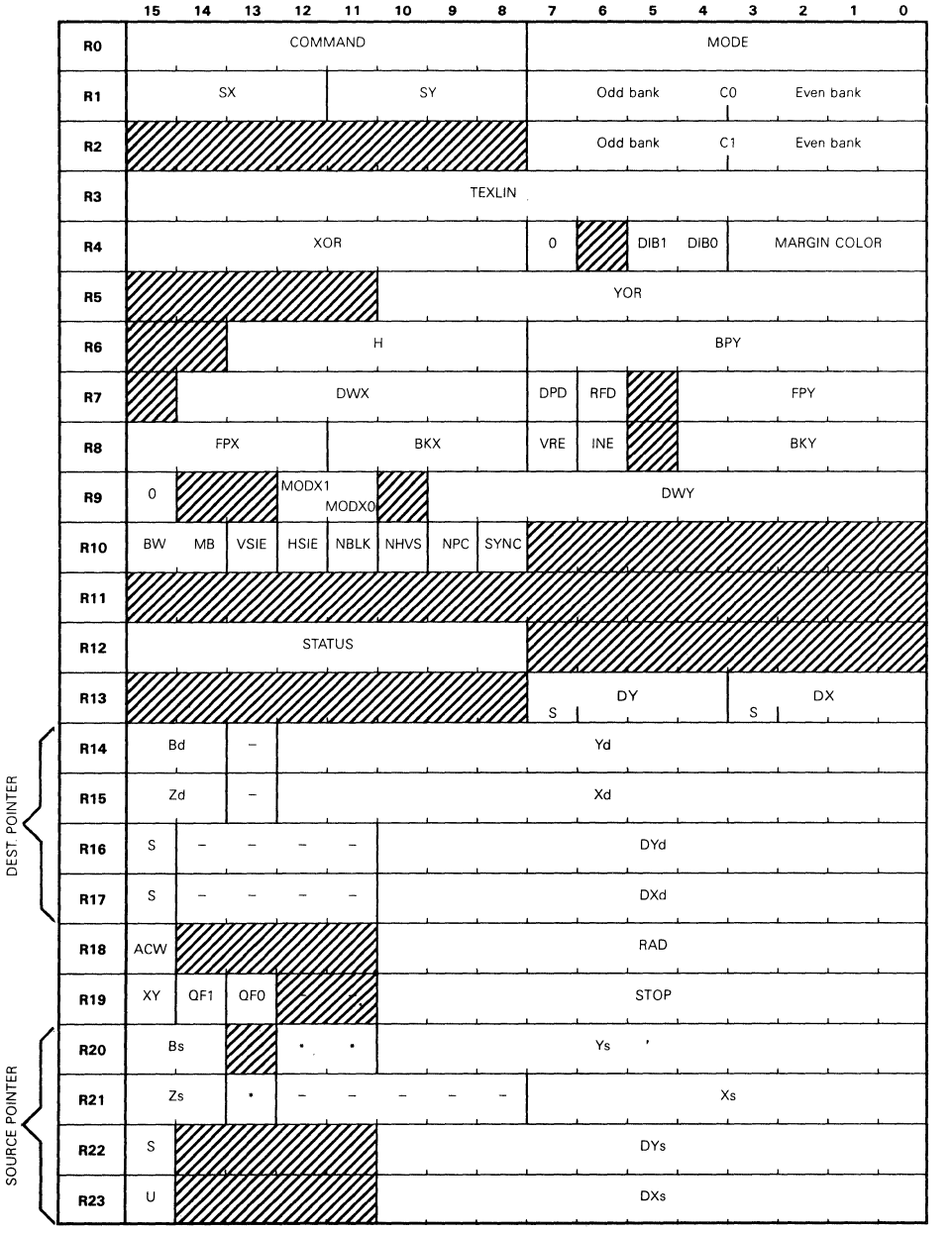
ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 V ± 5%, V_{SS} = 0, T_A = T_L to T_H) (unless otherwise specified)

CHARACTERISTIC	SYMBOL	MIN	TYP	MAX	UNIT
Supply voltage	V_{CC}	4.75	5	5.25	V
Input low voltage	V_{IL}	-0.3	–	0.8	V
Input high voltage	V_{IH}	2	–	V_{CC}	V
Input leakage current	I_{in}	–	–	10	µA
Output high voltage (I_{load} = -500 µA)	V_{OH}	2.4	–	–	V
Ouput low voltage I_{load} = 4 mA; ADM(0:15) I_{load} = 1 mA; other outputs	V_{OL}	–	–	0.4	V
Power dissipation	P_D	–	800	–	mW
Input capacitance	C_{in}	–	–	15	pF
Three state (off state) input current	I_{TSI}	–	–	10	µA

SECTION 7

7.1. REGISTER MAP



*: Don't care.
–: Used or not, according to the command.

7.2. COMMAND TABLE

TYPE			MNEM	CODE			PARAMETERS					ARGUMENTS							POINTERS								END COMMAND CURSOR POSITION		EXECUTION TIME		
				7	6	5	4	3	2	1	0	R0	R1	R2	R3	R13	R18	R19	R14	R15	R16	R17	R20	R21	R22	R23			INIT	LOOP	Per
DRAWINGS	LINEAR	DOT LINE	DLI	0	0	0	0	0	DMU	SP	SRU	X	X	X	X	X			X	X	X	X					Xd+DXd	Yd+DYd	5T	4T	DOT
		PEN LINE	PLI	0	0	0	POL	PEN	DMU	SP	SRU	X	X	X	X	X			X	X	X	X	X	X	X	X	Xd+DXd	Yd+DYd	5T	CELL+4T	CELL
		DOT ARC	DAR	0	0	1	0	0	DMU	SP	SRU	X	X	X	X	X	X	X	X	X	X	X					XF	YF	15T	10T	DOT
		PEN ARC	PAR	0	0	1	POL	PEN	DMU	SP	SRU	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	XF	YF	15T	CELL+10T	CELL
	AREA	RECTANGLE	REC	1	1	1	1	0	DMU	SP	SRU	X	X	X		X			X	X	X	X					Xd	Yd+DYd	10T	4T	AREA MEMORY WORD
		TRAPEZIUM	TRA	0	1	0	1	0	DMU	SP	SRU	X	X	X		X			X	X	X	X		X			Xd+DXd	Yd+DYd	10T	4T (NOTE 1)	
		POLYGON	FLL	0	1	0	0	BEG	DMU	SP	SRU	X	X	X		X			X	X	X	X					Xd+DXd	Yd+DYd	10T		
		POLYARC	FLA	0	1	1	0	BEG	DMU	SP	SRU	X	X	X		X	X	X	X	X	X	X					XF	YF	15T		
	CELL	PRINT CHARACTER	PCA	1	0	1	1	REP	DMU	SP	SRU	X	X	X		X			X	X	X	X	X	X	X	X	Xd+DXd	Yd	4T	6T	MEMORY WORD
		PRINT OBJECT	PVS	1	0	0	SMU	REP	DMU	1	SRU	X	X			X			X	X	X	X	X	X	X	X	Xd+DXd	Yd			
			PVF	1	0	1	0	REP	DMU	1	SRU	X	X			X			X	X	X	X	X	X	X	X	Xd+DXd	Yd			
ACCESS		LOAD VIEWPORT	LDV	1	1	1	0	XFT	0	0	INC								X	X	X	X	X	X	X	X	Xs	Ys	2T	5T	MEMORY WORD
		SAVE VIEWPORT	SAV	1	1	1	0	XFT	0	1	INC								X	X	X	X	X	X	X	X	Xs	Ys	2T	4T	
		MODIFY VIEWPORT	RMV	1	1	1	0	XFT	1	0	INC								X	X	X	X	X	X	X	X	Xs	Ys	2T	10T	
CURSOR		UP-DOWN MOVE	UDM	1	1	0	0	0	1	DWN	SRU					X			X	X	X						Xd	Yd+DYd	3T		
		LEFT-RIGHT MOVE	LRM	1	1	0	1	LEF	0	0	SRU					X			X	X		X					Xd+DXd	Yd	3T		
		DIAGONAL MOVE	CDM	1	1	0	1	LEF	1	DWN	SRU					X			X	X	X	X					Xd+DXd	Yd+DYd	4T		
CONTROL		NO OPERATION	NOP	1	1	0	0	0	0	0	0																		1T		
		ABORT	BRT	1	1	1	1	1	1	1	1																		1T		

DMU = 1 : Destination mask use.
SP = 1 : Short pel; long pel when SP=0.
SRU = 1 : Short relative register use (R13).
PEN = 0 : The pen is a single pel.
PEN = 1 : POL=0: the pen is the character cell addressed by the source pointer.
POL = 1: the pen is the object associated with a source mask addressed by the source pointer.
BEG = 1 : Initiate a polygon or polyarc filling.
This parameter should be reset only when the second drawing is not identical to the first one (Ex: first polygon, then polyarc).
INC = 0 : The source pointer is not auto-incremented.
INC = 1 : XFT=1: the source pointer is auto-incremented, X direction first.
XFT = 0: The source pointer is auto-incremented or auto-decremented, Y direction first.
REP = 1 : The cell is stepped and repeated through the destination window.
When REP=0, only one cell is printed.
SMU = 1 : The source mask is used.
DWN = 1 : The cursor is moved down (up if DWN=0).
LEF = 1 : The cursor is moved left (right if LEF=0).

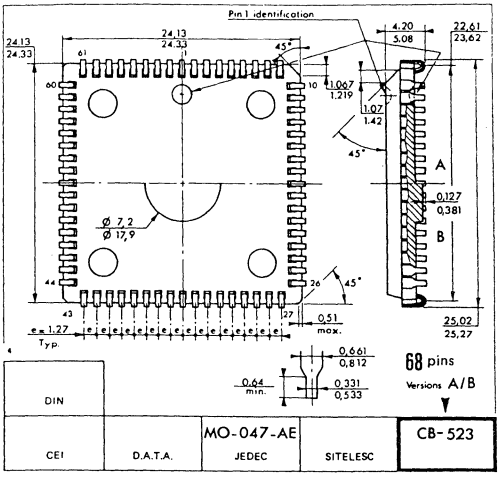
NOTE: With PVF command, any pel with color different from 0 has its source mask implicitly set and used. In other words, pels with color value 0 are transparent.
- DXd, DYd, and DYs are signed values.
- DXs is always positive.
- T = memory cycle = 8 CLK clock periods.
- For execution time, add to the short pel loop in the table:
· 1T if DMU=1
· 1T if SMU=1
· 2T if long pen are used
· 2T if mask printing is required.

Command execution is performed only out of the display periods.
NOTE 1: for FLL and FLA commands, add 4T and 8T respectively per pel belonging to the boundary.

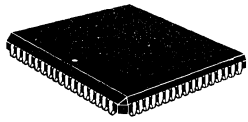
SECTION 8

MECHANICAL DATA AND ORDERING INFORMATION

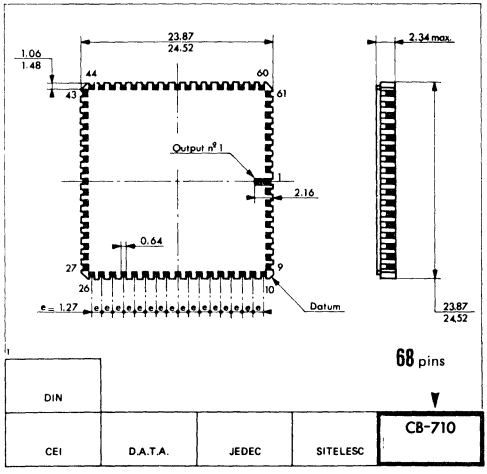
8.1. MECHANICAL DATA



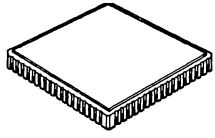
CB-523



FN SUFFIX
PLCC68



CB-710



E SUFFIX
LCCC68

8.2. ORDERING INFORMATION

PACKAGE TYPE	TEMPERATURE RANGE	PART NUMBER
PLCC FN Suffix	0° C to + 70° C	TS68483CFN
LCCC E Suffix	− 40° C to + 85° C − 55° C to + 125° C	TS684483VE TS 684483ME

THOMSON COMPONENTS MOSTEK

TS68494

PALETTE AND SERIAL PORT CONTROLLER

ADVANCE INFORMATION

HMOS 2

The TS68494 single chip color palette can be used in any low to mid range color graphics application to provide up to 256 colors freely selected between 4096 values.

Moreover, the TS68494 integrates a serial port controller for dual port video-RAM.

Powerful and cost effective graphics and text application may be achieved with THOMSON-SEMICONDUCTEURS Graphics processor family (TS68483/68493) and TS68494 chip set.

Main features :

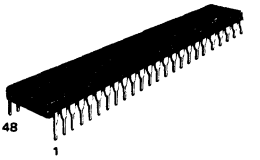
- One chip color look-up table
- 256 colors 8 bits/pixel) between 4096
- Up to 35 Mpixels/sec
- Three on-chip 4-bit video DACs
- RS-343A compatible video outputs
- Serial port controller for dual port video-RAM
- Logical operation between pixel value and programmable masks
- Horizontal zoom and panning capability
- Standard microprocessor interface, TS68000 compatible
- Single 5V supply
- 44 pin PLCC and 48 DIL packages

CASES

CB-521

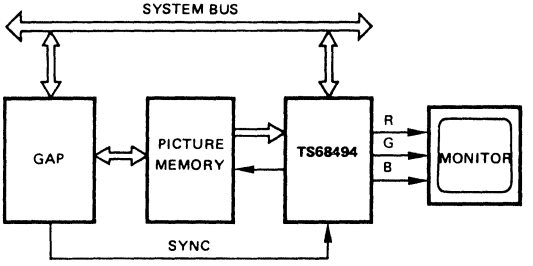
PLCC44

CB-229

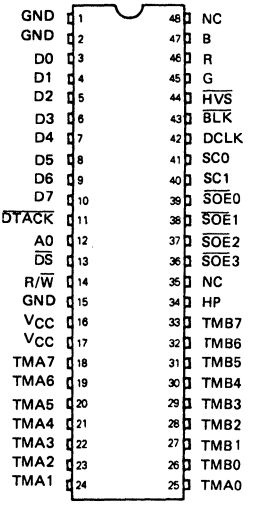


PLASTIC PACKAGE

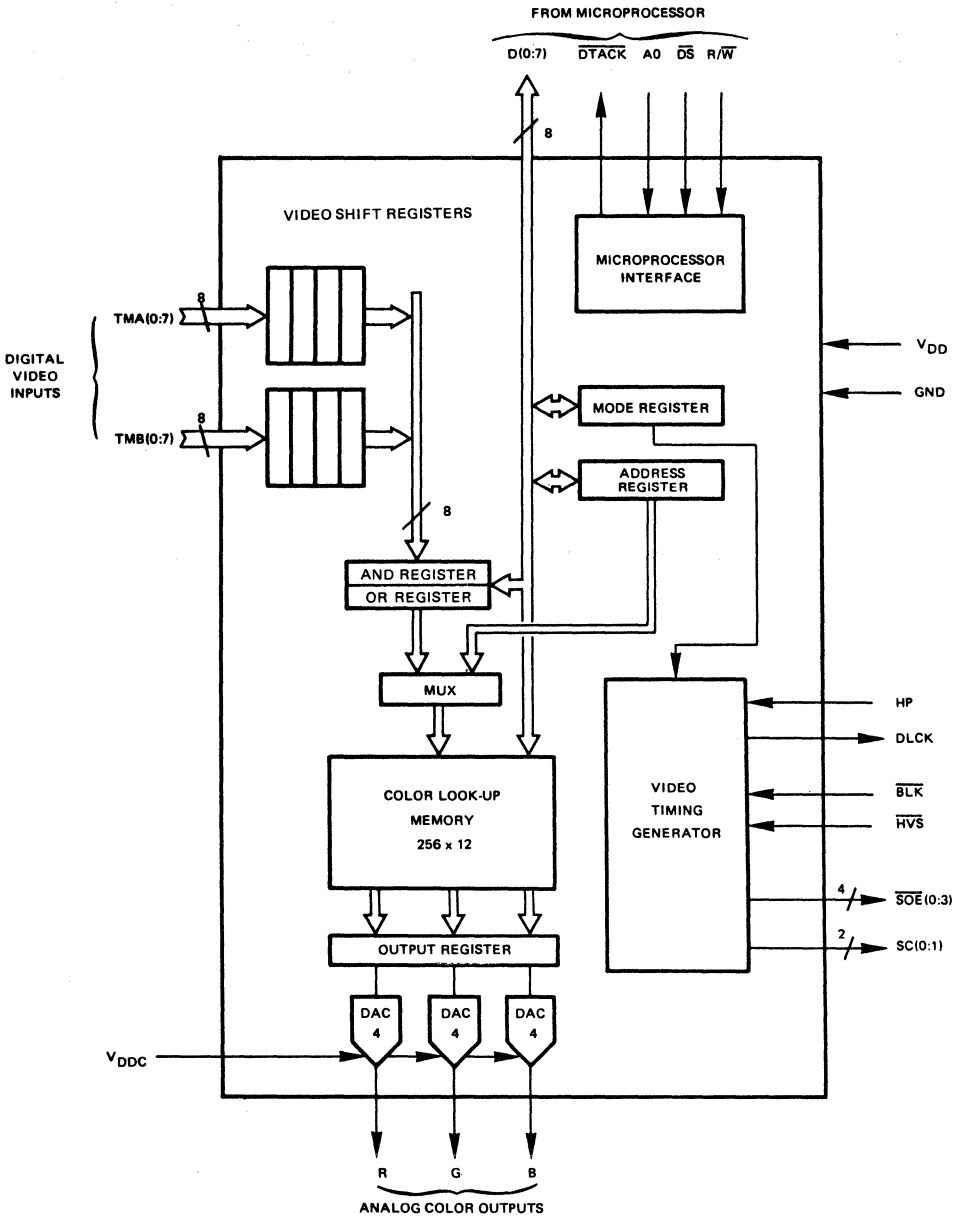
TYPICAL APPLICATION



PIN ASSIGNMENT



BLOCK DIAGRAM



FRAME BUFFER INTERFACE

Name	Pin type	No. PLCC	No. DIP	Function	Description
SC0 SC1	O	38 37	41 40	Serial port Shift clock	These two clocks, with inverted phase, shift data out of video-RAM serial ports during the active display period. Shift rate can be programmed at HP/8, HP/4 or HP/2 according to the Mode Register programming.
$\overline{SOE0}$ $\overline{SOE1}$ $\overline{SOE2}$ $\overline{SOE3}$	O	36 35 34 33	39 38 37 36	Serial output Enable	These signals are used to control the video-RAM serial port outputs in order to multiplex one to four outputs to the same bus.
TMA(0:7) TMB(0:7)	I	23-16 24-31	25-18 26-33	Pixel value Inputs	- In non multiplexed mode the 8 bit pixel value is fed to TMA (4 : 7) , (Respectively pixel bit 6, 4, 2, 0) and TMB (4 : 7) (Respectively pixel bit 7, 5, 3, 1) at pixel clock rate (HP) - In multiplexed mode, TMA(0:7) and TMB(0:7) are multiplexed in order to load the internal shift registers. During each cycle of eight HP clock periods, these sixteen lines are multiplexed four times to provide the value of eight pixels.

VIDEO INTERFACE

Name	Pin type	No. PLCC	No. DIP	Function	Description
HP	I	32	34	Pixel clock	This clock controls the internal shift registers operation and the internal Timing Generator.
DCLK	O	39	42	HP/8	This clock is the pixel clock HP divided by eight. It can be used as a basic clock by a Video Timing Generator of a graphics processor.
$\overline{BLK}$	I	40	43	Blanking	This active low input forces the R,G,B analog output to blank level.
$\overline{HVS}$	I	41	44	Composite synch.	This input is sampled by the rising edge of the DCLK clock.
R G B	O	43 42 44	46 45 47	Color output	These three analog outputs deliver the color signals from the internal DACs. The synch. information is delivered on the Green signal (RS-343A standard).

MICROPROCESSOR INTERFACE

Name	Pin type	No. PLCC	No. DIP	Function	Description
D(0:7)	I/O	3-10	3-10	Data bus	These eight bidirectionnal lines are connected to the microprocessor data bus.
A0	I	12	12	Address	During a microprocessor access, this line selects the higher byte (A0 = 0) or the lower byte (A0 = 1) of a register.
$\overline{DS}$	I	13	13	Data Strobe	This active low input selects the TS68494 for a microprocessor access.
R/$\overline{W}$	I	14	14	Read/Write	This control signal determines whether the TS68494 is read (R/$\overline{W}$ = 1) or written (R/$\overline{W}$ = 0).
$\overline{DTACK}$	O	11	11	Data transfer Acknowledge	This active-low open drain output signals the completion of a bus cycle

OTHER PINS

Name	Pin type	No. PLCC	No. DIP	Function	Description
V_{CC}	S	15	16-17	Power supply	+ 5 V.
V_{DDC}	S	1		Analog power supply	Power supply for the internal DACs. + 5 V for RS-343A standard.
GND	S	2	1-2-15	Ground	

NOTE : Pin No. 35 in DIP package in not connected.

FUNCTIONAL DESCRIPTION

MICROPROCESSOR INTERFACE

The TS68494 is directly compatible with TS68000 microprocessor family and can be easily interfaced with most of the other standard microprocessor families such as 6809, 8086 (Figure 1).

Data transfer between the TS68494 and the microprocessor is made by logical 16-bit words through an 8-bit data bus. The microprocessor has access to seven internal registers.

Write access

During a write operation, a 16-bit word contains a 12-bit data field and 3-bit address field. The address field selects the register to be accessed (Figure 2).

As the higher byte (A0 = 0) contains the address, it must be loaded first.

Data written to the color register is loaded into the color look-up memory at the location pointed by the Color Address Pointer (CAP). When bit IW is set in the Mode Register, the CAP is automatically incremented after each access to the lower byte (A0 = 1).

Read access

Before any read operation, the Read Address Register (RAR) must be initialized.

When the contents of RAR is different from 0, the register pointed by RAR can be read by the microprocessor at address A0 = 1.

When RAR = 0, the color look-up memory location pointed by the Color Address Pointer (CAP) can be read by the microprocessor. Each 12-bit word of the color look-up memory can be read at address A0 = 0 and A0 = 1. When bit IR is set in the mode register, the CAP is auto-incremented after each access to address A0 = 1, thus allowing sequential access to the color look-up memory.

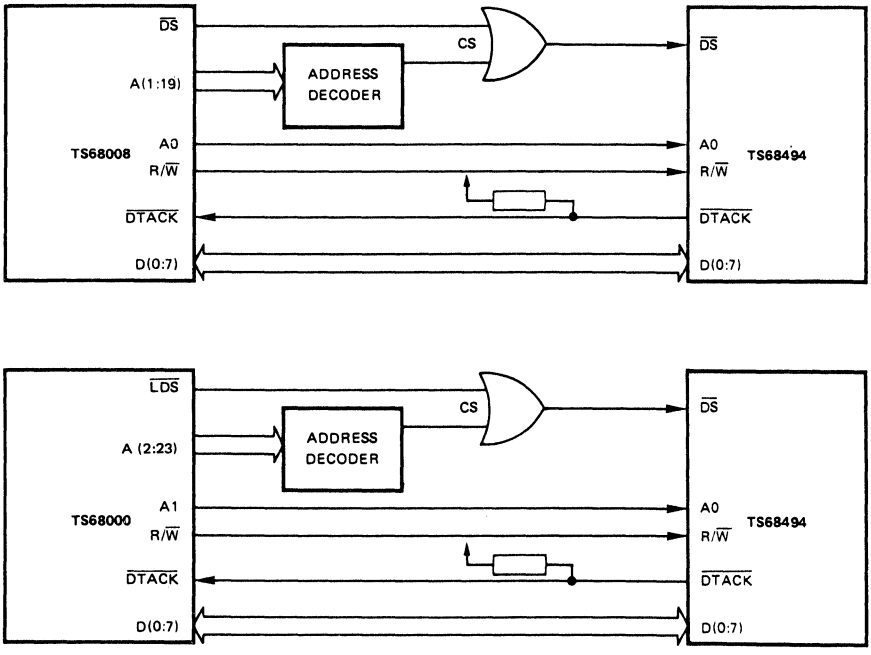
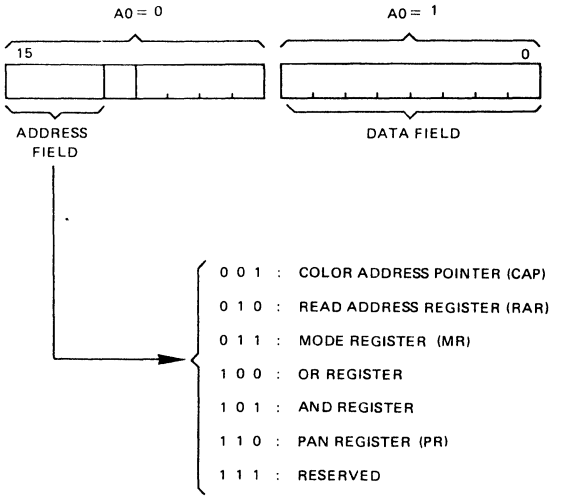
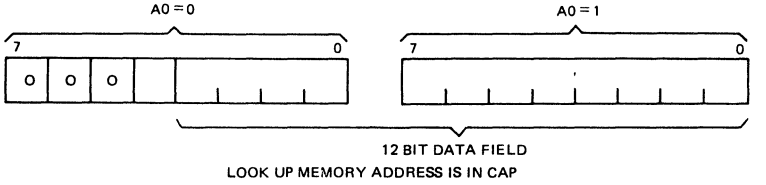


FIGURE 1 – INTERFACE WITH TS68000 FAMILY

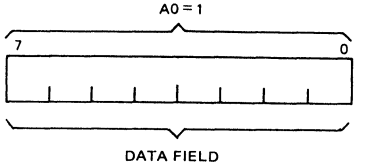
REGISTER ADDRESSING – Write operation



COLOR REGISTER ADDRESSING – Write operation



REGISTER ADDRESSING –Read operation



COLOR REGISTER ADDRESSING – Read operation

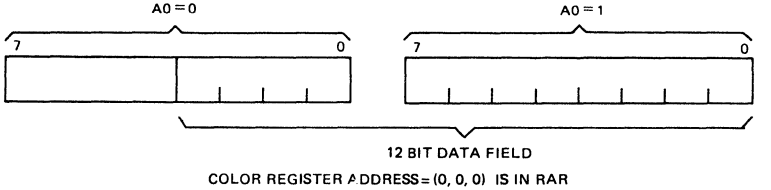
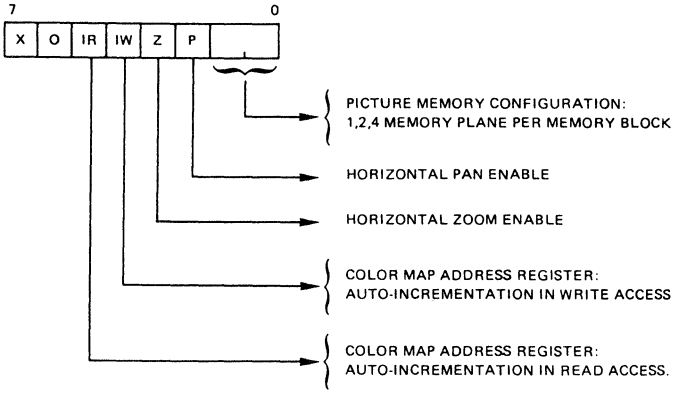


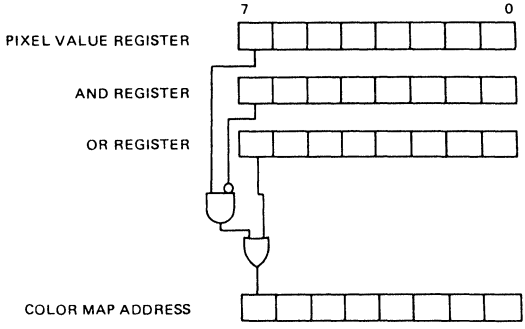
FIGURE 2 – REGISTER ADDRESSING

REGISTER DESCRIPTION

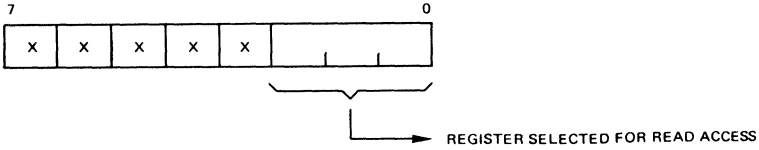
MODE REGISTER-MR



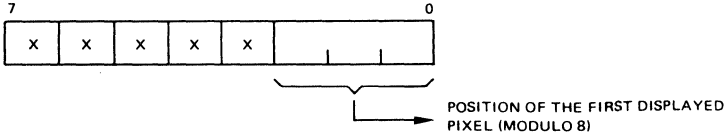
AND/OR REGISTERS



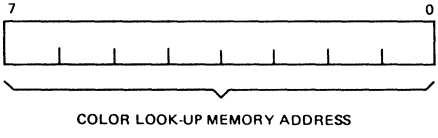
READ ADDRESS REGISTER-RAR



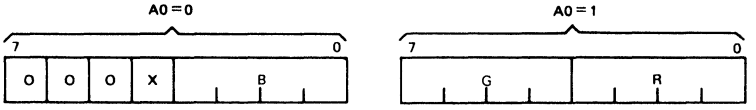
PAN REGISTER-PR



COLOR ADDRESS POINTER-CAP



COLOR REGISTER



VIDEO PROCESS

The pixel value received by the TS68494 is logically ANDed and ORed with the AND and OR registers. The result is used to address the color look-up memory.

The 12-bit word at the output of the memory is latched into the color output register. Each 4-bit field is fed to the respective DAC which delivers the analog signal conforming to RS-343A standard (Figure 3).

Zoom operation

The zoom mode is active when bit MR3 is set in the mode register. Except for the DCLK signal, all the clocks generated by the TS68494 Timing Generator are divided by two.

Pan operation (MR2 = 1)

This mode allows horizontal scrolling from 0 to 7 pixels, through the Pan Register PR.

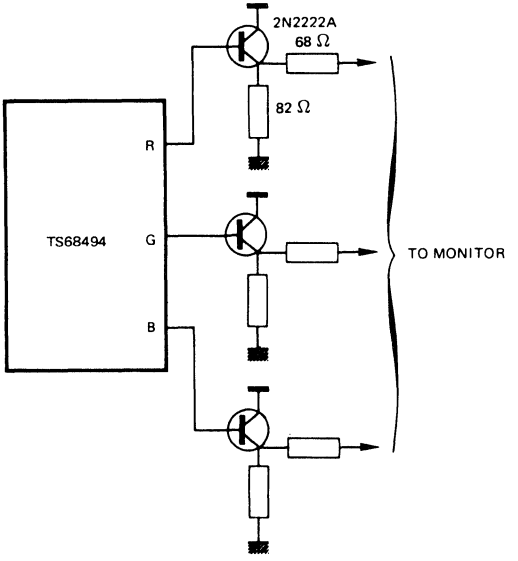


FIGURE 3 – RS343-A COMPATIBLE VIDEO INTERFACE

OPERATING MODES

The TS68494 provides four different operating modes, depending on the frame buffer organization.

Non multiplexed mode (MR1 = 1, MR0 = 1)

In this mode, the 8-bit pixel value present on TMA(4:7) and TMB(4:7) inputs is latched on the rising edge of HP clock. After AND and OR operations, it addresses the color look-up memory (Figure 8).

Memory organization

With the three multiplexed modes, the frame buffer must be organized into blocks of video-RAMs. Each block provides eight serial ports outputs, which deliver after each shift operation the information for eight consecutive pixels, one bit/pixel.

According to the memory type used and the size of the screen, a memory block may contain one, two or four memory bit planes. For example, a block made of two 64K x 4 video-RAMs can contains one plane of 1024 x 512, two planes of 512 x 512 or four planes 512 x 256 (Figure 4).

The timing generator outputs SC (0 : 1), SOE (0 : 3) are enabled by the BLK signal.

Multiplexed mode 0 (MR1 = 0, MR0 = 0)

In this mode, each memory block contains only one memory bit plane. The serial outputs of the four memory blocks holding the even planes are multiplexed to TMA(0:7) lines. Those of the memory blocks holding the odd planes are multiplexed to TMB(0:7) lines (Figure 5).

During each cycle of eight pixel clock periods, the TMA and TMB inputs are multiplexed four times. The multiplexing process is achieved through the Shift Output Enable signals $\overline{SOE}$(0:3) provided by the TS69494. The Serial Shift Clocks for the video-RAM serials ports SC0 and SC1 are provided by the TS68494 by dividing the pixel clock by eight.

Multiplexed mode 1 (MR1 = 0, MR0 = 1)

Each memory block contains two memory bit planes. Only $\overline{SOE}$0 and $\overline{SOE}$1 are used to multiplex two blocks. The shift clocks SC0 and SC1 delivered by the TS68494 are obtained by dividing the pixel clock by four (Figure 6).

Multiplexed mode 2 (MR1 = 0, MR0 = 0)

Each memory block contains four memory bit planes. Only $\overline{SOE}$0 signal is used. The Shift Clock SC0 is obtained by dividing the pixel clock by two (Figure 7).

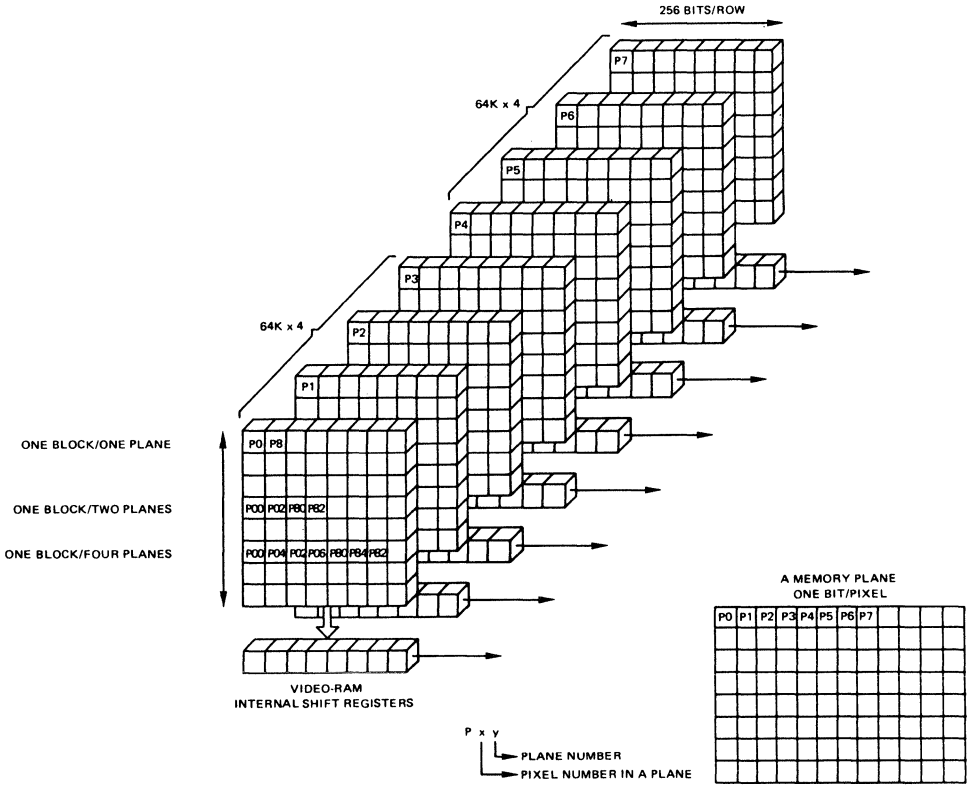
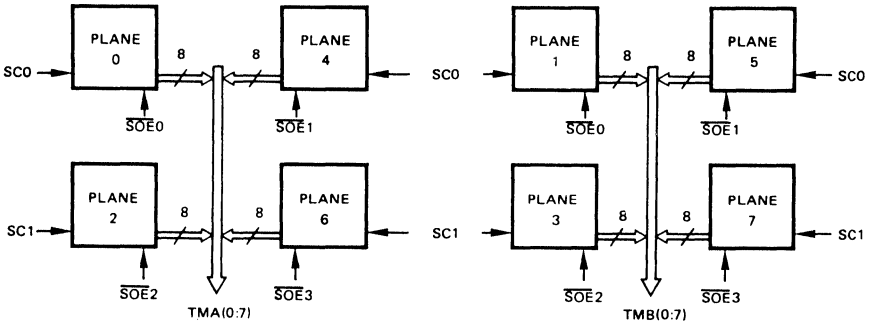


FIGURE 4 – MEMORY BLOCK ORGANIZATION



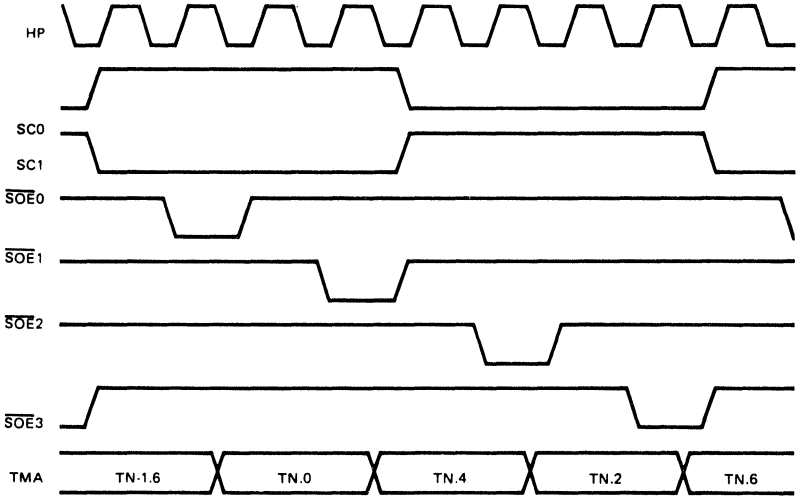
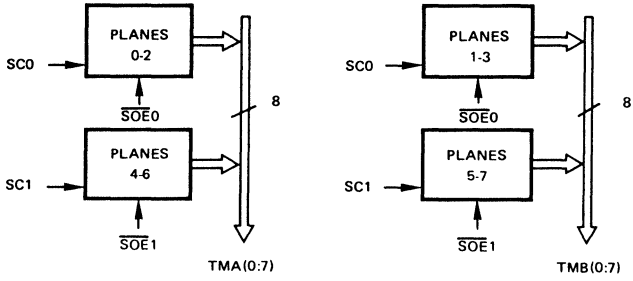


FIGURE 5 – MULTIPLEXED MODE 0



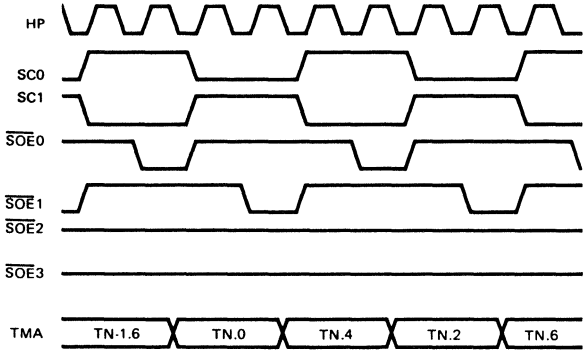
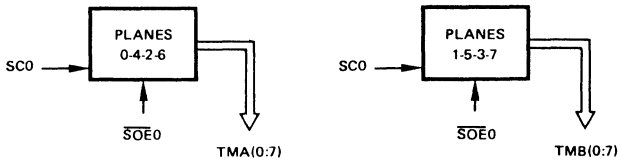


FIGURE 6 – MULTIPLEXED MODE 1



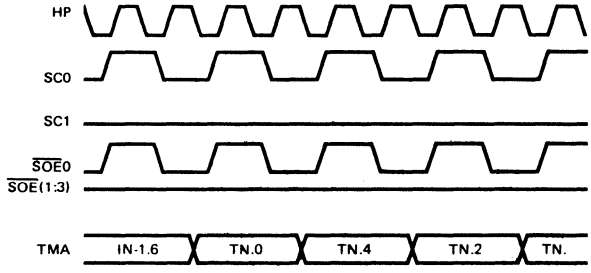


FIGURE 7 – MULTIPLEXED MODE 2

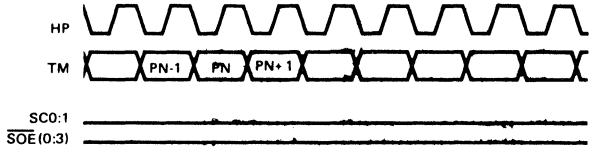


FIGURE 8 – NON MULTIPLEXED MODE

MAXIMUM RATINGS

Stresses above those hereby listed may cause permanent damage to the device. The ratings are stress ones only and functional operation of the device at these or any conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. Standard MOS circuits handling procedure should be used to avoid possible damage to the device.

Rating	Symbol	Value	Unit
Supply voltage	V_{CC}*	+0.3 to +7.0	V
Input voltage	V_{in}*	−0.3 to +7.0	V
Operating temperature range	T_A	0 to +70	°C
Storage temperature range	T_{stg}	−55 to +150	°C
Maximum power dissipation	P_{Dm}	+1.5	W

* With respect to GND

ELECTRICAL OPERATING CHARACTERISTICS

(V_{CC} = 5.0 V ± 5 %. V_{SS} = 0. T_A = 0 to + 70°C) (Unless otherwise specified)

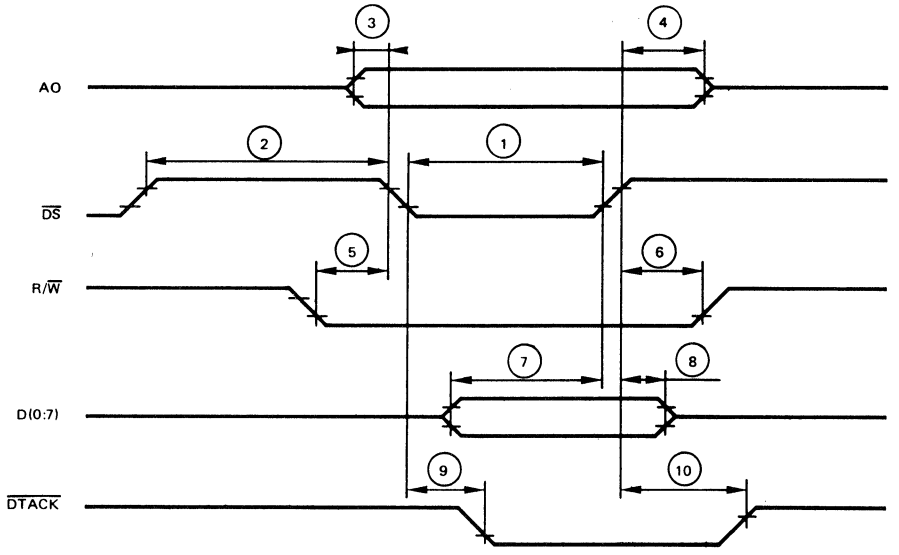
Characteristic	Symbol	Min	Typ	Max	Unit
Supply voltage	V_{CC}	4.75	5	5.25	V
Analog supply voltage	V_{DDC}	–	5	–	V
Input low voltage	V_{IL}	0.3	–	0.8	V
Input high voltage	V_{IH}	2	–	V_{CC}	V
Input leakage current	V_{IO}	–	–	10	μA
Output high voltage (I_{load} = − 500 μA)	V_{OH}	2.4	–	–	V
Output low voltage I_{load} = 4 mA	V_{OL}	–	–	0.4	V
Power dissipation	P_D	–	700	–	mW
Input capacitance	C_{in}	–	–	15	pF
Three state (off state) input current	I_{TSI}	–	–	10	μA

MICROPROCESSOR INTERFACE

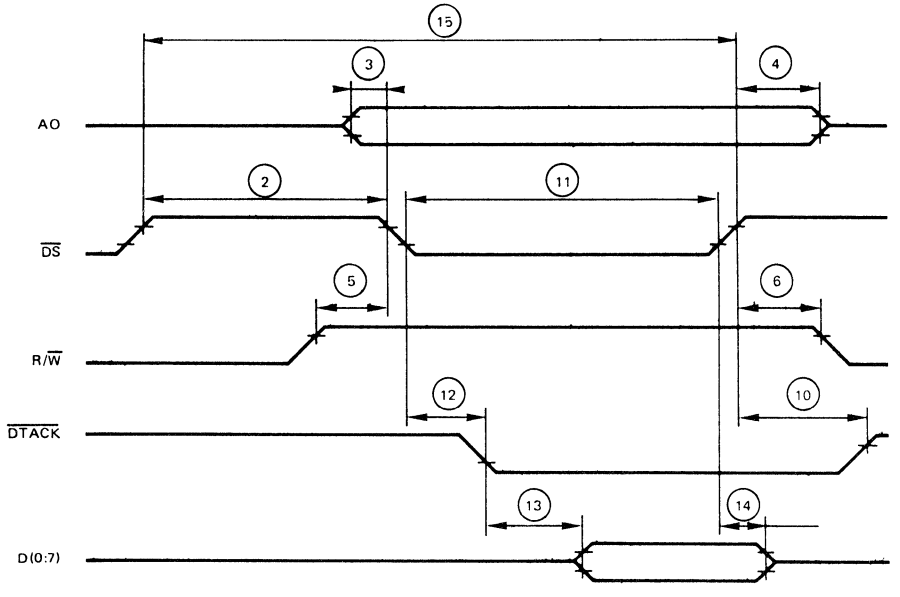
Microprocessor interface timing:
V_{CC} = 5.0 V ± 5 %. T_A = 0° to 70° C. C_L = 100 pF on D(0:7) , CL = 130pF and RL = 500Ω on $\overline{DTACK}$
Reference levels: V_{IL} = 0.8 V and V_{IH} = 2 V on all inputs.
V_{OL} = 0.4 V and V_{OH} = 2.4 V on all outputs

No.	Characteristic	Min.	Max.	Unit
1	$\overline{DS}$ low (write cycle)	60	–	ns
2	$\overline{DS}$ high	60	–	ns
3	A0 set up time to $\overline{DS}$	0	–	ns
4	A0 hold time from $\overline{DS}$	10	–	ns
5	R/$\overline{W}$ set up time to $\overline{DS}$	0	–	ns
6	R/$\overline{W}$ hold time from $\overline{DS}$	10	–	ns
7	Data set up time (write)	50	–	ns
8	Data hold time (read)	0	–	ns
9	$\overline{DTACK}$ delay from $\overline{DS}$	50 + PP(6THP −(2))	–	ns
10	$\overline{DTACK}$ high from $\overline{DS}$	110	–	ns
11	$\overline{DS}$ low (read cycle)	120	–	ns
12	$\overline{DTACK}$ low from $\overline{DS}$ (read)	100+5 HP	–	ns
13	Data delay from $\overline{DTACK}$ (read)	–	30	–
14	Data hold time from $\overline{DS}$	10	–	ns
15	Cycle time	70+8 HP		

TIMING DIAGRAM 1 – MICROPROCESSOR INTERFACE



WRITE CYCLE



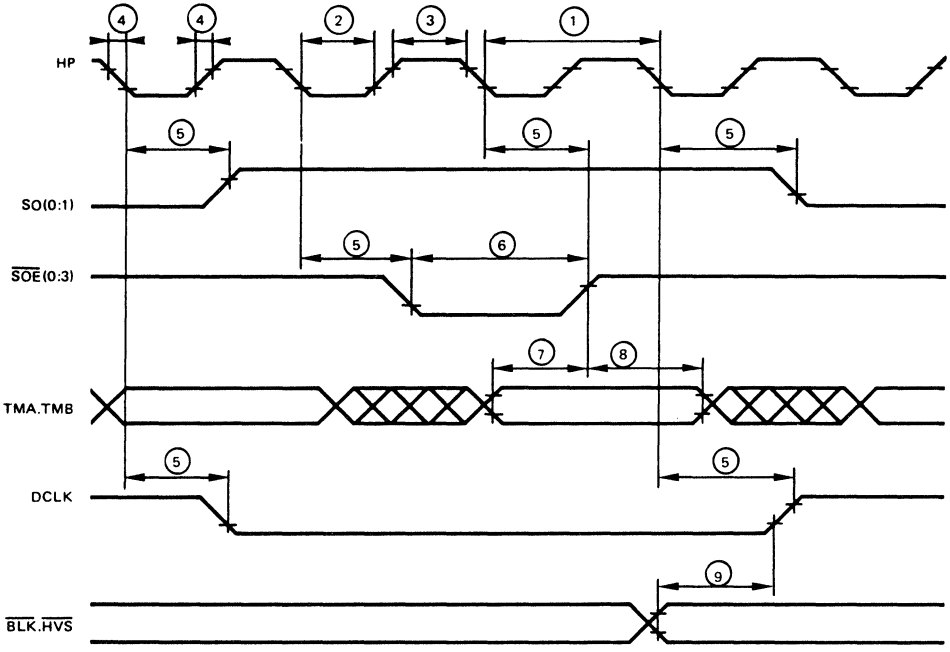
READ CYCLE

DIGITAL VIDEO SIGNALS

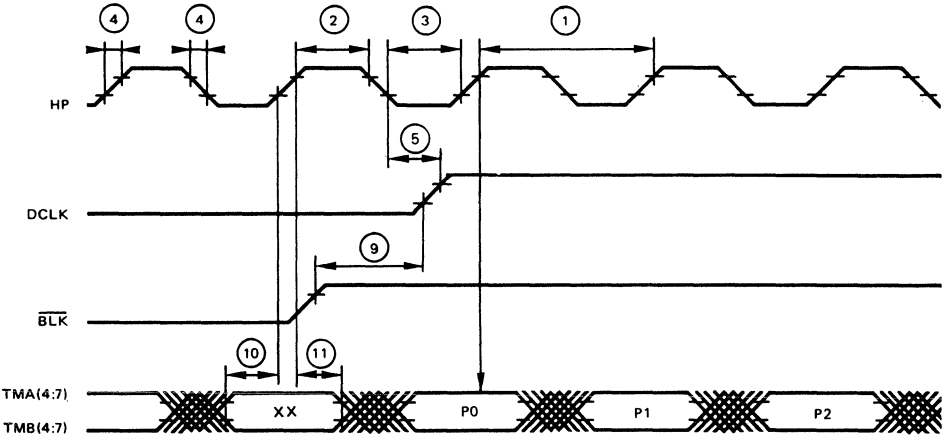
V_{CC} = 5.0 ± 5 %. T_A = 0° to + 70° C. C_L = **50 pF** on DCLK, SC(0:1), $\overline{SOE}$(0:3)
Reference levels: V_{IL} = 0.8 V and V_{IH} = 2 V on all inputs.
V_{OL} = 0.4 V and V_{OH} = 2.4 V on all outputs.

No.	Characteristic	Min.	Max.	Unit
1	HP clock period	33	125	ns
2	HP high pulse width	10	–	ns
3	HP low pulse width	10	–	ns
4	HP rise and fall time	–	5	ns
5	Timing generator output delay from HP	–	28	ns
6	$\overline{SOE}$ low pulse width	THP	–	ns
7	$\overline{BLK}$, $\overline{HVS}$ set up time to DCLK	60	–	ns
8	TMA, TMB set up time to $\overline{SOE}$	-2	–	ns
9	TMA, TMB hold time from $\overline{SOE}$	12	–	ns
10	TMA(4:7), TMB(4:7) set up time to HP	5	–	ns
11	TMA(4:7), TMB(4:7) hold time from HP	5	–	ns

TIMING DIAGRAM 2 -- DIGITAL VIDEO SIGNALS



MULTIPLEXED MODES 0, 1, 2,

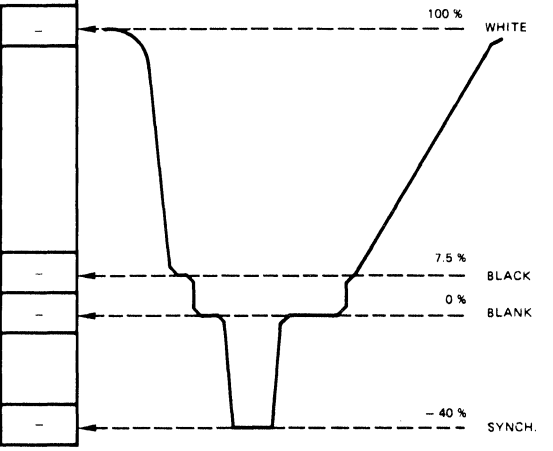


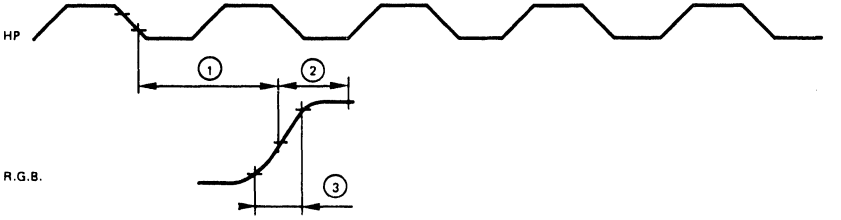
NON MULTIPLEXED MODE 3

ANALOG VIDEO OUTPUTS

V_{DDC} = 5 V, T_A = 0° to + 70°C

R.B			G		
MIN	TYP	MAX	MIN	TYP	MAX
–	2.2	–	–	2.77	–
–	0.88	–	–	1.46	–
–	0.76	–	–	1.35	–
–	–	–	–	0.76	–

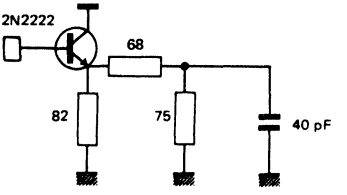




TIMING DIAGRAM 3

No.	Characteristic	Min.	Max.	Unit
1	delay from HP (50 %)	–	30	ns
2	1/2 LSB settling	–	15	ns
3	Rise and fall time (10 % - 90 %)	–	10	ns

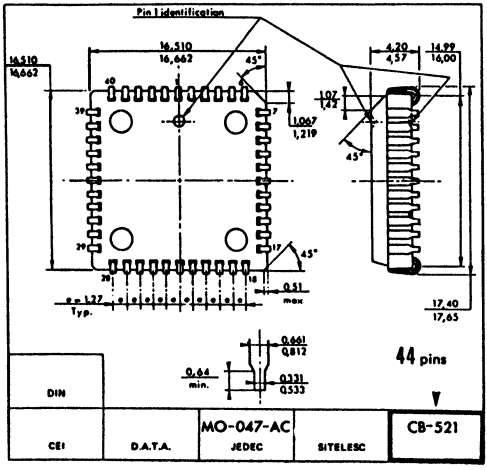
Test load

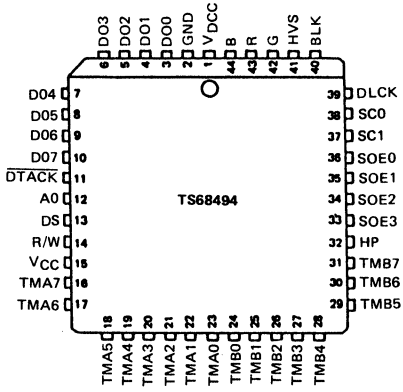


PHYSICAL DIMENSIONS

CB-521

PLCC44





CHAPTER 7 - DIGITAL SIGNAL PROCESSING ICs

DIGITAL SIGNAL PROCESSING ICs SELECTION GUIDE

Part number	Description	Technology	Number of pins	Page
TS68930	High speed general purpose digital signal and arithmetic processor with on-chip RAM ROM, multiplier, alu. accumulators and I/O	HMOS	48	7-3
TS68931	ROMless version of TS68930	HMOS	84	7-3
TS68950	Modem transmit analog interface	CMOS	28	7-55
TS68951	Modem receive analog interface	CMOS	28	7-69
TS68952	Modem transmit/receive clock generator	CMOS	28	7-93

THOMSON COMPONENTS

MOSTEK

TS68930 • TS68931

PROGRAMMABLE SIGNAL PROCESSOR

ADVANCE INFORMATION

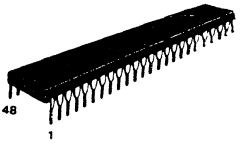
The TS68930/1 (Programmable Signal Processor) is a high-speed general purpose signal and arithmetic processor with on-chip memory, multiplier, ALU, accumulators and I/Os. It is organized in a parallel/pipeline structure to execute simultaneously one ALU, function, multiplication, two reads and one write operation and associated address calculation every 160 ns.

- Parallel/pipeline Harvard architecture
- 3 data-bus structure
- 3 data types : 16-bit real, 32-bit real
  : 16 + 16-bit complex number
- 2 versions : TS68930 (internal ROMs) 48-pin
  : TS68931 (external ROMs) 84-pin
- Pipeline complex multiplier
- 2 x 128 x 16-bit RAM
- 512 x 16-bit coefficient ROM
- 32-bit instruction bus
- 64 k x 32-bit external program space
- 68000 family compatibility
- Dual external buses : local/system

HMOS2

PROGRAMMABLE SIGNAL PROCESSOR

CASE CB-229



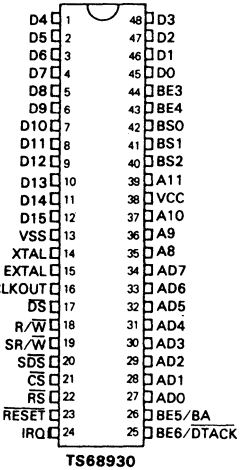
TS68930
P SUFFIX
PLASTIC PACKAGE

TS68931
E SUFFIX
LCCC 84

TYPICAL APPLICATIONS

- Adaptive processing
- Complex numbers
- Digital filtering
- Fast Fourier transform
- Voice grade communication systems
- High-speed modems
- Speech processing
- Audio Frequencies
- Sonar/radar
- Image processing
- Robotics
- Graphics processing

PIN ASSIGNMENT

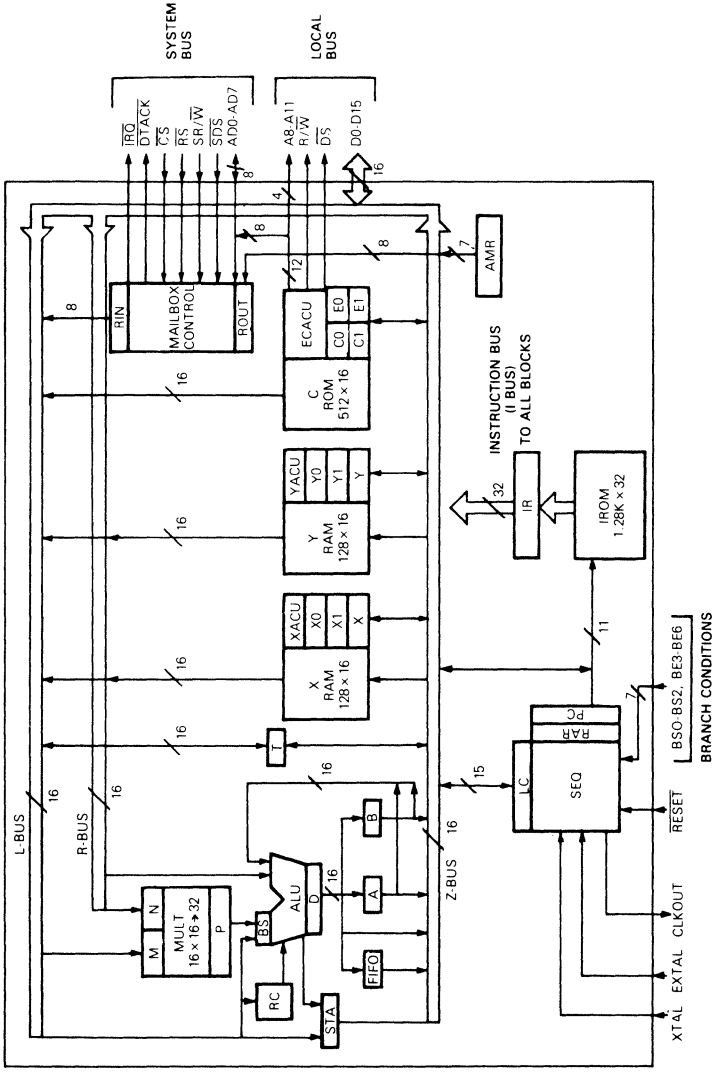


TS68930

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

SECTION 1
BLOCK DIAGRAM



SECTION 2
PIN DESCRIPTION

LOCAL INTERFACE

Name	Pin Type	Pin nb. TS68930	Pin nb. TS68931	Function	Description
D (0:15)	I/O	45-48 1-11	6-21	Data bus	Can be concatenated or separate D (0:7), D (8:15)
A (8:11)	O	35,37,39	45-48	Address bus	High order addresses for local interface
$\overline{DS}$	O	17	27	Data Strobe	Synchronizes the transfer
R/$\overline{W}$	O	18	28	Read/write	Indicates the current bus cycle state
CLKOUT	O	16	26	Clock output	The frequency of CLKOUT is one half the frequency of the input clock or crystal

SYSTEM INTERFACE

Name	Pin Type	Pin nb. TS68930	Pin nb. TS68931	Function	Description
AD (0:7)	I/O	27-34	35-42	System data bus or local address bus	The data exchanges between the processor and a master via a mailbox is the function of this bus. It is also used to generate the addresses of an external RAM.
$\overline{CS}$	I	21	31	Chip Select	Used by a master to gain access to the mailbox and system bus
$\overline{RS}$	I	22	32	Register Select	Used by a master to gain access to the mailbox and system bus
$\overline{SDS}$	I	20	30	System Data Strobe	Synchronizes the transfer on the system bus
SR/$\overline{W}$	I	19	29	System Read/Write	Indicates the current system bus cycle state
$\overline{DTACK}$	O	25	43	Data Transfer Acknowledge	Indicates that the processor has recognized it is being accessed
BA	O	26	44	Bus Available	Indicates availability of system bus to master
$\overline{IRQ}$	O	24	34	Interrupt Request	Handshake signal sent to the master to gain access to the mailbox

EXTERNAL BRANCH CONDITIONS

Name	Pin Type	Pin nb. TS68930	Pin nb. TS68931	Function	Description
BS (0:2)	I	42-40	49-51	Branch on State	External conditions. Can be programmed on a high or low state
BE (3:6)	I	44 43 26 25	53 52 44 43	Branch on Edge	External conditions. Falling edge is memorised and reset when tested. BE5 shares pin with $\overline{BA}$ BE6 shares pin with $\overline{DTACK}$

OTHER PINS

Name	Pin Type	Pin nb. TS68930	Pin nb. TS68931	Function	Description
EXTAL	I	15	25	Clock	Crystal input pin for internal oscillator or input pin for external oscillator
XTAL	I	14	24	Clock	Together with EXTAL it is used for the external 25 MHz crystal
VDD	I	38	23-65	Power supply	
VSS	I	13	22-64	Ground	
$\overline{RESET}$	I	23	33	Reset	

INSTRUCTION INTERFACE (TS68931 only)

Name	Pin Type	Pin nb.	Function	Description
I (0:31)	I/O	1-5 56-63 56-84	Instruction Address/data bus Coefficient ROM address bus	Instruction bus - 32-bit data Instruction address - (I16 - I31) External coefficient ROM address - (I6 - I15) 10-bit (9-bit address + output enable signal)
$\overline{HALT}$	I	54	Halt Signal	Halts the processor. This signal freezes the program counter and loop counter
INCYCLE	O	55	Instruction cycle clock	160 ns in REAL mode 320 ns in CPLX/DBPR Mode

SECTION 3
SUMMARY OF BASIC HARDWARE

OPERATING MODES

Resource	Paragraph N°	Symbol	Function
Mode register	5.1	MODE	2-bit register defining the operating mode (real/complex/double precision).

CONTROL BLOCK

Resource	Paragraph N°	Symbol	Function
Instruction ROM	5-2-1	IROM	1280 × 32-bit word read-only-memory containing program code and immediate data.
Instruction register		IR	32-bit register containing instruction.
Program Counter	5-2-2	PC	16-bit register containing address of program memory.
Sequencer	5-2-3	SEQ	The sequencer can test directly 16 conditions programmed on a high or low state.
Return Address Register	5-2-4	RAR	16-bit register for saving program counter in case of subroutine call.
Loop Counter	5-2-5	LC	15-bit register containing a control word for automatic loop. It is divided as follows.
		LCI	4-bit register containing the number of instructions to be executed in the loop.
		LCR	8-bit register containing the number of loops.
		LCD	3-bit register containing the number of instructions between declaration and start of the loop.

PROCESSING BLOCK

Resource	Paragraph N°	Symbol	Function
Pipeline Multiplier	5-3-1	MULT	16 × 16 → 32 parallel pipeline multiplier + 16-bit adder/substractor to execute complex multiplications.
		M, N	2 × 16-bit registers containing multiplier operands.
		P	2 × 16-bit register containing multiplier result.
Barrel Shifter	5-3-2	BS	Variable 0 - 15-bit right shift, left shift, right rotation barrel shifter.
Arithmetic Logic Unit	5-3-3	ALU	2 port 16-bit arithmetic logic unit. 5 possible sources; 4 possible destinations; 27-functions. Works on 32-bit in 2 cycles.
		D	ALU output register.
Saturation	5-3-4	SAT	Flag. Indicates saturation mode
Status	5-3-5	STA	15-bit register containing status of ALU, mode, status of address calculation units
Accumulators	5-3-6	A	2 × 16-bit accumulator.
		B	2 × 16-bit accumulator.
Fifo	5-3-7	F	4 × 16-bit first in first out register.
Empty Fifo		EF	Flag.Indicates that the fifo is empty ; can be set by software.
Replace Code register	5-3-8	RC	6-bit register allowing replacement of ALU operation code by a data coming from L-BUS.
Transfer register	5-3-9	T	2 × 16-bit register providing direct transfer between L-BUS and Z-BUS.

MEMORY BLOCK

Resource	Paragraph N°	Symbol	Function
Data RAMs	5-4-1	XRAM YRAM	2 × 128 × 16-bit word random access memories containing data.
Data ROM		CROM	512 × 16-bit word read only memory containing coefficients or constants.
Address Calculation Units	5-4-3	XACU YACU	2 × 7-bit arithmetic units providing incrementation, decrementation, automatic loop of address. XACU is dedicated to XRAM. YACU is dedicated to YRAM.
		ECACU	12-bit arithmetic unit providing incrementation, decrementation of address. Shared between CROM and ERAM (external RAM).
Pointers	5-4-4	X0, X1 X	2 × 7-bit registers used for indirect addressing of XRAM. Supplementary register used for circular addressing.
		Y0, Y1 Y	2 × 7-bit registers used for indirect addressing for YRAM. Supplementary register used for circular addressing.
		C0, C1	2 × 9-bit registers used for indirect addressing of CROM.
		E0, E1	2 × 12-bit registers used for indirect addressing of ERAM.
XRAM Circular Flag	5-4-5	XC	Flag. Indicates the circular addressing mode for XRAM.
YRAM Circular Flag		YC	Flag. Indicates the circular addressing mode for YRAM.

INPUT/OUTPUT BLOCK

Resource	Paragraph N°	Symbol	Function
Access Mode Register	5-5	AMR	7-bit register defining the access mode on the 2 external buses (local and system).
Input Register	6-4	RIN	3 × 8-bit shift register. Mailbox input.
Output Register		ROUT	3 × 8-bit shift register. Mailbox output.
Ready Out Internal	6-5	RDYOIN	Flag used in the protocol to indicate which processor has access to the mailbox.

SECTION 4
ARCHITECTURE

4.1. INTERNAL ARCHITECTURE

4.1.1. Parallel processing

The processor internal architecture is organized around the following blocks :
- the arithmetic logic unit and its associated working registers
- the multiplier
- the 3 memories and their associated address calculation units
- the transfer register
- the I/O unit.

All these blocks can work simultaneously and independently.

4.1.2. Three-bus structure

To avoid memory access bottlenecks the processor architecture includes 3 data buses. Two read buses (L-BUS and R-BUS) continuously feed the operating units. Thus making it possible to load the ALU and the multiplier with the two operands simultaneously. The write bus (Z-BUS) is used to transfer the results back into the RAMs (internal or external).

4.1.3. Wide instruction word

The 32-bit wide instruction format allows the processor to execute the following operations in 1 instruction cycle:
- Read two operands (from internal or external memories)
- Execute an ALU operation
- Start a multiplication
- Use the result of the multiplication started 2 cycles before
- Write a result in internal/external memory
- Post-modify 3 pointers independently
- Store data into the transfer register.

4.1.4. Pipeline (cf. fig. 4.1.)

The figure 4.1. outlines the overlap of the instruction prefetch and execution as well as the pipelined data operation.

By using a pipeline structure, the processor performs efficiently on all digital signal processing algorithms. For example the result of a multiplication started at instruction IN will be available at IN + 2. That will not prevent from starting a new multiplication at IN + 1 which in turn will be available at IN + 3, etc... in effect, giving a multiplier throughput of 1 multiplication every cycle.

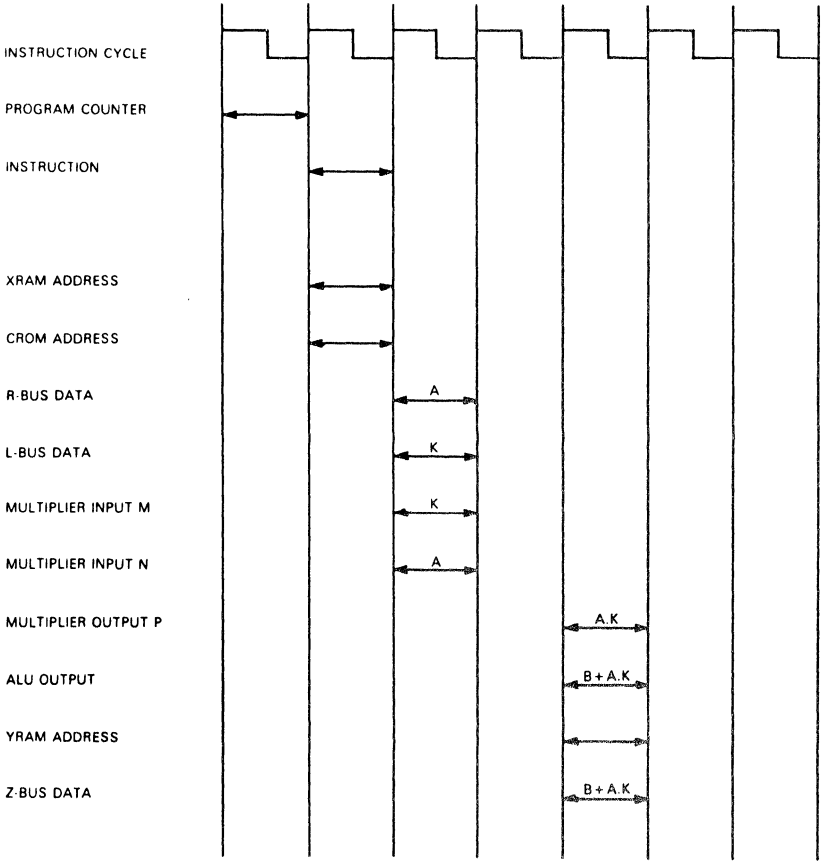


FIGURE 4.1.A. - PIPELINE DELAY

Example: READ A(XRAM), READ K(CROM), MULTIPLY A and K, ADD B(ACCUMULATOR), WRITE RESULT A.K + B INTO YRAM.

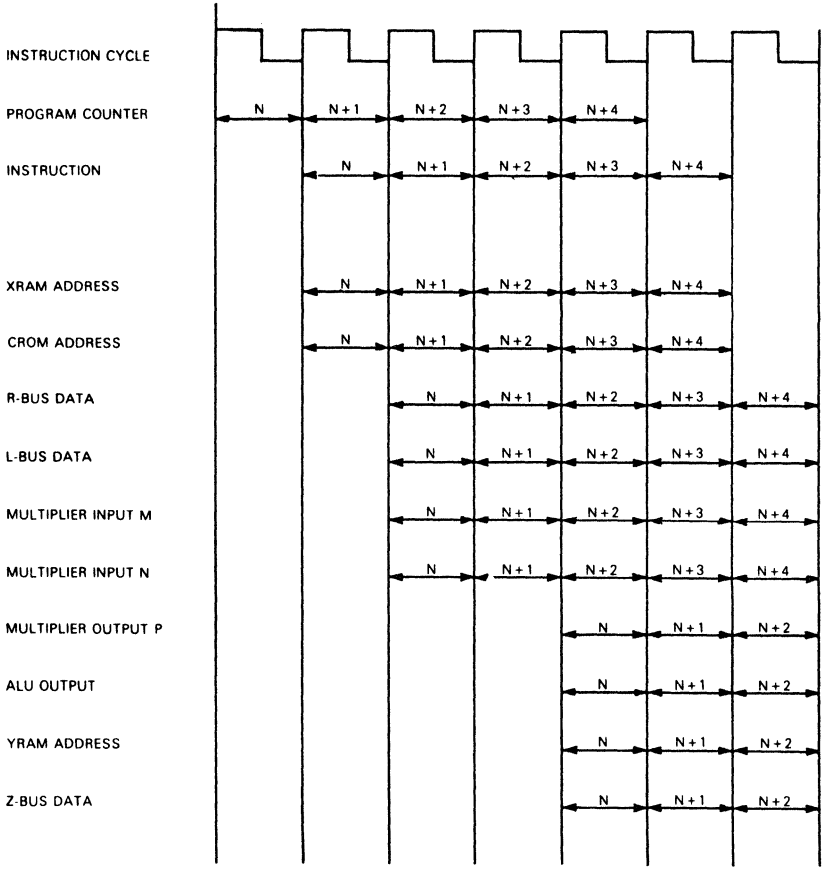
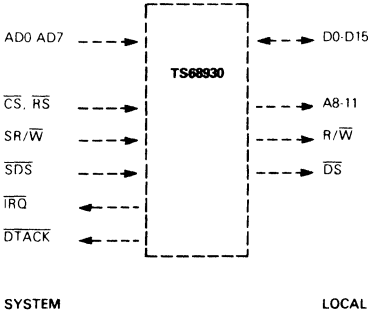


FIGURE 4.1.B. - PIPELINE THROUGHPUT

Example: The operation A.K + B (described in fig. 4.1.A.) is executed every cycle.

4.2. EXTERNAL ARCHITECTURE

The TS68930 is provided with two external buses:
— the system bus: AD0-AD7
— the local bus: D0-D15.
The processor is a slave on the system bus.
The processor is a master on the local bus.



SYSTEM BUS

The main use of the system bus is for the processor to exchange information with a general purpose microprocessor or another TS68930 in a multiprocessor environment.

The informations are exchanged through a mailbox with a flag ($\overline{IRQ}$) indicating to the master (the other processor) that it can gain access to the mailbox.

LOCAL BUS

The main use of the local bus is for the processor to exchange information with an external memory, a peripheral, a data converter or another TS68930 in a multiprocessor environment. All these external circuits are defined as slaves.

The processor is the master of its local bus, i.e it generates the address and control signals which direct the exchange on the local bus. This bus is a direct extension of the internal structure and all external circuits connected on it, work in exactly the same way as the internal operating units.

SECTION 5
FUNCTIONAL DESCRIPTION

5.1. OPERATING MODES

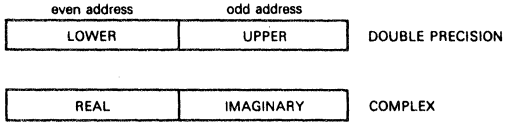
The processor provides three operating modes set by programming, each mode representing a different data type :
- REAL = 16-bit data
- COMPLEX (CPLX) = 2 x 16-bit data
- DOUBLE PRECISION (DBPR) = 32-bit data.

The modes are made transparent to the programmer as all operating units and all working registers are provided with the right length.

Main differences between real mode and complex or double precision mode:

a) In complex and double precision mode the memory space is reduced by half as all operands are 32-bit long (cf. format below).

b) The instruction cycle time is doubled (320 ns instead of 160 ns) as all operations are made sequentially.



5.2. CONTROL BLOCK

5.2.1. Instruction ROM: IROM

The instruction ROM has a capacity of 1280 x 32-bit in the MCU version. It can be extended to 64 K x 32-bit in the MPU version.

5.2.2. Program counter: PC

The program counter is 16-bit wide, 11 bits are used in the MCU version.

5.2.3. Sequencer: SEQ

The sequencer increments the program counter except in case of sequence jump which are listed below:
a) immediate branch
b) computed branch
c) jump to subroutine } (cf 5.2.4.)
d) return from subroutine }
e) automatic loop } (cf 5.2.5.)

In case of immediate branch the PC is loaded with an immediate value whereas in case of computed branch the PC is loaded with a value coming from the accumulators (A, B), the FIFO (F) or the transfer register (T).

The sequencer can test directly 16 conditions programmed on a high or low state:

BRANCH NEVER/ALWAYS

STATUS CONDITIONS

- SR Sign (Real)
- SI Sign (Imaginary)
- CR Carry (Real)
- CI Carry (Imaginary)
- Z Zero
- OVF Overflow
- MOVF Memorized overflow

The memorized overflow (MOVF) is reset when tested by the branch instruction.

EXTERNAL CONDITIONS

- BS0-BS2 } External pins
- BE3-BE6 }

The falling edges of BE3-BE6 are memorized internally and reset when tested by the branch instruction.

The external test conditions are used to synchronise different processes or as a ready input flag in multiprocessor system.

MAILBOX FLAG

- RDYOIN Internal mailbox flag

5.2.4. Return address register : RAR

The JSR instruction allows one level of subroutine nesting with automatic saving of the PC on to the return address register (RAR).

Multiple level of subroutine nesting can be implemented in RAM using either of the two pointers as stack pointer.

In this case the RAR is used as the last level of nesting.

5.2.5. Loop counter: LC

a) The efficiency of executing repeated calculations often encountered in Digital Signal Processing is considerably improved by using the loop counter since the instructions for counter increment and range check are no longer needed.

This counter can implement a loop of up to 16 instructions repeated 256 times with a delay of up to 8 instructions.

b) **DESCRIPTION:**

LCI : Instruction Loop Counter : 4-bit
Counts the number of instructions to be executed in the loop

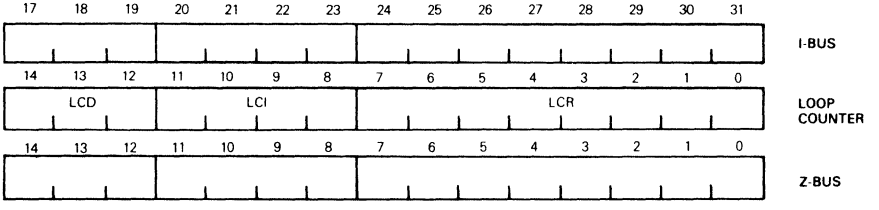
LCR : Repeat Loop Counter: 8-bit.
Gives the number of times the loop will be repeated.

LCD : Delay Loop Counter: 3-bit.
Gives the delay between the declaration and the start of the loop.

c) **USE:**

A loop is declared by loading the instruction loop counter and the delay loop counter with a constant (INI Instruction) and LCR with a constant or a variable (INI or OPDI instruction).

The loop counter contents can be saved (SVR instruction) with the following format:



Asserting $\overline{\text{HALT}}$ will freeze the state of the LC.

Asserting $\overline{\text{RESET}}$ will reset the LC.

5.3. PROCESSING BLOCK

5.3.1. Multiplier

a) The multiplier executes a 16 x 16 → 32-bit signed multiplication every instruction cycle with a delay of 2 cycles independently of the operating mode.

The number representation is signed 2's complement and the result format for the 3 modes is shown in figure 5.3.1.

b) **USE:**

The multiplier is always active. To start a multiplication the two operands are loaded into the two input registers(M, N).

The multiplication will be repeated every cycle until one or both operands are changed. The processor offers the possibility of loading the two input registers independently.

The result is available in the product register (P) two cycles later.

c) **COMPLEX MULTIPLICATION:**

The processor executes a complex multiplication:
$(A+jB) \cdot (C+jD) = AC - BD + j(AD+BC)$
every 320 ns thanks to an internal 80 ns clock.

As it can be seen from the equation the complex multiplication can generate an overflow. In this case the multiplier overflow (OVFM) is memorised inside the status register.

d) **NOTES:**

No provision is made for the operation 8000 x 8000 (hexadecimal).
If this condition arises the product will be 8000 (hexadecimal).

After changing modes the product P is calculated following the new mode.

The signal HALT (cf. Input/output) will inhibit the loading of the product register P.

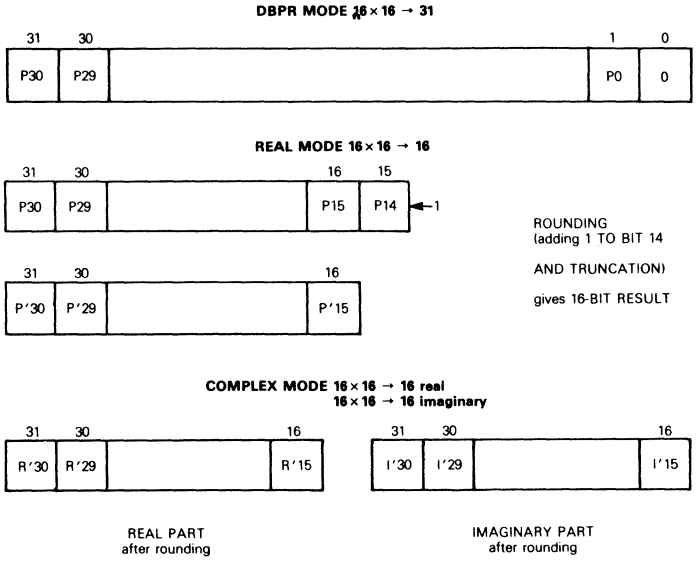


FIGURE 5.3.1. - MULTIPLICATION OUTPUT REGISTER (P) FORMATS

5.3.2. Barrel Shifter (BS)

All shift and rotation operations are performed at the L-side (left) ALU input. The operand can come from two sources:
- LBUS
- P (product register)

There are two types of shift and rotate operations:

1) The operations which are part of the ALU code:
- arithmetic shift right by 1 (ASR)
- logical shift right by 1 (LSR)
- arithmetic shift left by 1 (ASL)
- logical shift left by 1 (LSL)
- logical shift right by 8 (LSRB)
- logical shift left by 8 (LSLB)
- rotate right by 1 (ROR)

2) The operations which are implemented through dedicated instructions:
- ASR (0 → 15) arithmetic shift right by N $0 \leqslant N \leqslant 15$
- LSR (0 → 15) logical shift right by N
- LSL (0 → 15) logical shift left by N
- ROR (0 → 15) rotation right by N

Note:
In double precision the shift operations are not executed on 32 bits, but on 2 x 16-bit as the barrel shifter is a 16-bit unit. In complex mode the shift operations are executed on the real and imaginary parts.

5.3.3 ALU

The ALU inputs are called L-Side (Left) and R-Side (Right).

There are two possible sources on the L-Side:
- L BUS
- P (multiplier output).

There are two possible sources on the R-Side:
- R BUS
- Accumulators A or B.

The selection between A or B is made by the field ALU destination (refer to operating codes). If the ALU destination field is B then the ALU source is B. In all other cases A will be used.

The ALU output is called D.

There are four possible destinations for D:
- Accumulator A
- Accumulator B
- FIFO
- Z-BUS (no working registers are modified).

ALU CODES
There are 27 ALU codes. The list is shown in figure 7.9.

5.3.4. Saturation mode (SAT)

If the saturation mode is set (SAT flag) the circuit will behave as follows :
- Positive overflow = ALU result is forced to 7FFF (hexadecimal)
- Negative overflow= ALU result is forced to 8000 (hexadecimal)

The saturation mode does not apply to the double precision mode.

5.3.5. Status register: STA

a) DESCRIPTION

15	14	13	12	11	10	9	8	7	6	5	4 3	2	1	0
SR	SI	CR	CI	Z	OVF	MOVF	AOVF	OVFM	EF	SAT	MODE	XC	YC	–

CONDITION CODE REGISTER (CCR):

SR	Sign (real)	Set if the msb of the ALU result is 1. Cleared otherwise.
SI	Sign (imaginary)	Set if the msb of ALU imaginary result is 1. Cleared otherwise.
CR	Carry (real)	Set if a carry is generated out of the msb of the operand for arithmetic and shift operations. Cleared otherwise.
CI	Carry (imaginary)	Set if a carry is generated out of the msb of the imaginary part for arithmetic and shift operations. Cleared otherwise.
Z	Zero	Set if the result equals zero. In complex mode it is equivalent to the imaginary and real parts being both zeros.
OVF	Overflow	Set if there was an arithmetic overflow. This implies that the result is not representable in the operand size. In complex mode it is equivalent to the overflow of the imaginary or real part.
MOVF	Memorised overflow	Set as overflow. Reset when tested by a branch instruction.
AOVF	Advanced overflow	Exclusive or of bit 14 and bit 15 of the ALU. Set if there was an arithmetic overflow on half capacity (15 bits in real/complex mode, 31 bits in double precision mode). Cleared otherwise.
OVFM	Overflow (Multiplier)	Set if the multiplier adder/substractor has overflowed. Only meaningful for complex multiplication. Cleared otherwise.

STATE REGISTER

EF	Empty FIFO	Set if the FIFO is empty. Cleared otherwise.
SAT	Saturation mode flag	Set if the PSI is in saturation mode. Cleared otherwise.
MODE (2 bits)	Operating mode	Real, complex or double precision.
XC	XRAM	Circular addressing mode flag.
YC	YRAM	Circular addressing mode flag.

b) USE

The status can be saved (instruction SVR).

The condition code register can be read (in OPIN instruction) and it can be loaded from a RAM via L-BUS (ALU code LCCR) without passing through the ALU.

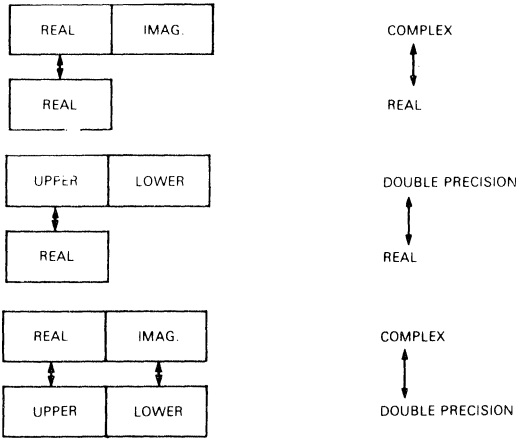
The state register can be programmed by an INI instruction.

5.3.6. Accumulators: A,B

a) The processor provides two distinct accumulators (A and B). In real mode they are 16-bit long. In complex and double precision mode they are 32-bit long.

b) Changing modes, changes the length of the accumulator and the relation between the words described below.

CHANGING MODES



It must be noted that the imaginary (respectively lower) part of the word remains unmodified when switching to real mode.

5.3.7. FIFO: F

a) **FUNCTION**

Highly pipelined algorithms require a series of pipeline registers between the ALU output and the memories in order to store intermediate results.

This is precisely the function of the 4 x 16-bit first-in first-out (FIFO) register.

b) **DESCRIPTION**

It is a 4 x 16-bit deep register that becomes 2 x 32-bit in complex and double precision modes (cf. format below).

c) **USE**

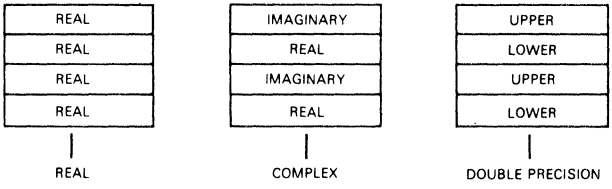
When the FIFO is full it becomes impossible to write into it.

When the FIFO is empty a status bit (EF) is set.

This bit can also be set by programmation.

d) **NOTE**

In real mode, a result loaded at instruction IN into an empty FIFO will be available for transfer to the RAM at IN + 2. In all other cases it will be at IN + 1.



5.3.8. Replace code register: RC

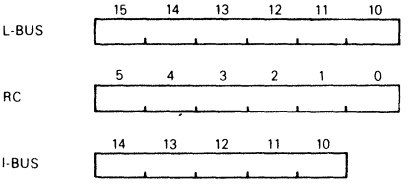
a) **FUNCTION**

The function of this register is to control the ALU by a data coming from the memories via L-BUS instead of an instruction. In other words it allows the data to take control of program sequencing without using test instructions.

For this reason it can be said that the **instructions are data controlled**.

b) **DESCRIPTION**

It is a 6-bit register with the following format :



BIT 1-5 = ALU code is substituted by this value

BIT 0 = 0 Destination of ALU output = accumulator A
= 1 Destination of ALU output = accumulator B

c) **USE**

This register is controlled by three ALU codes :

ALU code	Function
RCR	Load ALU control code in RC
RCE	Execute ALU control code contained in RC
RCER	Execute ALU control code contained in RC Load new ALU control code in RC

5.3.9. Transfer register: T

a) **FUNCTION**

It is a bidirectionnal register standing between L-BUS, and Z-BUS.

It can be a source and a destination to both buses.

Among its numerous uses, it can perform the function of :
- Loop back to the multiplier in one cycle
- Temporary register between memory and ALU
- Temporary register between memory and multiplier
- Operations between accumulators
- Memory to memory transfer.
- Saving program counter.

b) **DESCRIPTION**

It is a 16-bit register extended to 32 bits in complex and double precision mode.

c) **USE**

The relation between the 32-bit and the 16-bit word in case of mode switching is identical to the accumulators relation.

In branch instruction the register can be used to save the PC.

When the mode is complex the PC (16-bit) is saved into the real part of the register, when the mode is DBPR the PC is saved into the upper part of the register.

T can also be used as a source of the PC:

When the mode is complex the PC is loaded with the real part of the register, when the mode is DBPR the PC is loaded with the upper part of the register.

5.4. MEMORY BLOCK

5.4.1. Data memories : XRAM, YRAM, CROM

The processor architecture allows the connection of four memories:

• 2 internal RAMs
XRAM 128 x 16-bit
YRAM 128 x 16-bit

• 1 internal data ROM separated from the program ROM
CROM 512 x 16-bit
In the microprocessor version this ROM is external.

• 1 external memory
ERAM 4 K x 16-bit
This external memory is accessed in a single cycle (160 ns) in exactly the same way as the internal memories. Moreover it does not require any "glue" parts to be connected to the processor.

Notes :

1. In complex and double precision modes all data are 32-bit long. Hence the available memory space is divided by two.

2. The instruction set allows **any combinations of simultaneous use of these memories ;** the only restraints are :
   - Reading and writing in the same RAM in the same cycle.
   - Accessing CROM and ERAM simultaneously.

5.4.2. Addressing modes

The processor provides four addressing modes:
• Indirect addressing with post modification.
• Direct addressing.
• Immediate addressing.
• Circular addressing mode (also called virtual shift mode).

5.4.3. Address calculation units: ACU

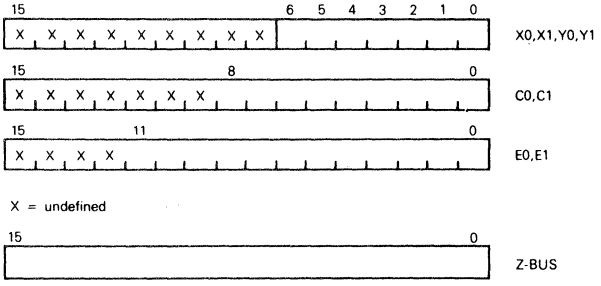
Combining these four addressing modes and the processor 3-bus structure implies the need to generate **at each instruction cycle three different addresses.** To realise these functions each memory is associated with an address calculation unit:
- XRAM with XACU
- YRAM with YACU
- CROM or ERAM with ECACU.

5.4.4. Pointers: X0, X1, Y0, Y1, C0, C1, E0, E1, X, Y

Indirect addressing is the most commonly used addressing mode in vector or signal processing. For this reason the processor offers a large number of pointers (10): X0, X1, Y0, Y1, C0, C1, E0, E1 + X and Y for circular mode.

Each memory can be addressed by two pointers and pointers can be increased (+1) decreased (−1) or held (+0) independently.

They can also be loaded with new addresses (constants or computed values) and saved in case of context switching (cf. format below).



5.4.5. Circular adressing mode

a) FUNCTION

This feature is used to simulate the function of a shift register without moving the data stored. It is particularly useful in filtering and convolution functions.

b) DESCRIPTION
X0 : lower limit
X1 : upper limit
X : current address
(respectively Y0, Y1, Y for YRAM)

The algorithm can be described as follows:

1. ADDRESS: ADDRESS + 1 (post-incrementation)
IF ADDRESS GREATER THAN UPPER LIMIT THEN ADDRESS = LOWER LIMIT

2. ADDRESS: ADDRESS − 1 (post-decrementation)
IF ADDRESS SMALLER THAN LOWER LIMIT THEN ADDRESS = UPPER LIMIT

c) USE

Programming the circular addressing mode is done independently of the operating modes (real, complex or double precision), in the following way. With reference to the instruction OPCODE: example XRAM.

1. Initialization instruction (INI)
Circular addressing bit set (K7 = 1)
Load X0 with lower limit.

2. Initialization instruction (INI)
Circular addressing bit set (K7 = 1)
Load X1 with upper limit.

3. INI or OPDI instruction
Load X with current address (a value between X0 and X1).

After the first instruction the circular addressing mode is effective.

From now on the programmer has access only to pointer X and X1. All instructions referencing pointer X0 will now physically reference pointer X.

To gain access again to pointer X0 the programmer goes back to the normal mode by an initialisation instruction.

d) FLAGS

When a RAM is in the circular addressing mode, a flag (XC, YC) is set inside the status.

5.4.6. ODD/EVEN addresses

a) In complex and double precision modes the processor automatically generates the two addresses of the word (even then odd).

	COMPLEX WORD	DBPR WORD
even address	real part	lower part
odd address	imaginary part	upper part

The processor offers the possibility to inverse this order by writing a 1 into the ADOF bit (refer to OPCODE).

ADOF

0 even followed by odd
1 odd followed by even.

b) **USE**

This feature is made available independently or simultaneously for XRAM and YRAM.

With reference to OPCODE.

XRAM

Initialization instruction (INI)
- select complex or double precision mode
- select pointer X0 or X1 and load it with J constant
- select ADOF bit as wanted (0 or 1).

YRAM

Initialization instruction (INI)
- select complex or double precision mode
- select pointer Y0 or Y1 and load it with K constant
- select ADOF bit as wanted (0 or 1).

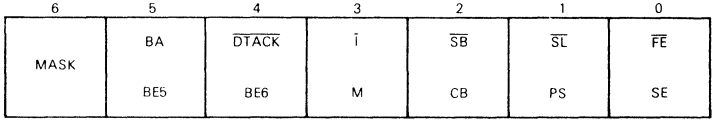
5.5. ACCESS MODE REGISTER: AMR

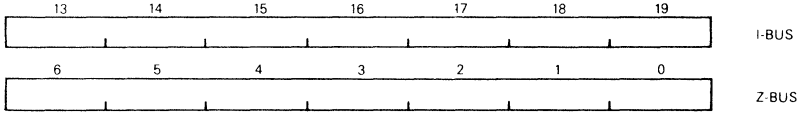
a) **DESCRIPTION**

This register defines the processor external access modes.
Its contents can be initialized with a constant and saved into memory, (cf. format below).

It is a 7-bit register each bit being defined as shown below:

$\overline{FE}$/SE :Fast exchange/slow exchange on local bus
$\overline{SL}$/PS :Slave/pseudo-Slave on system bus
$\overline{SB}$/CB :Concatenated or separate local bus
I/M :Local bus control signal types
$\overline{DTACK}$/BE6 :BE6 pin redefinition
BA/BE5 :BE5 pin redefinition
MASK :Allows the AMR to be masked by the external halt (microprocessor version only).





BIT 0: $\overline{FE}$/SE

0 FAST EXCHANGE = external access in 160 ns (1 cycle)

1 SLOW EXCHANGE = external access in 320 ns (2cycles).

The slow exchange mode:
- Can only be used in the real mode.
- The circuit automatically repeats the instruction which defines the external transfer.
- The control of the multiplier, ALU, ACUs, loop counter is the responsibility of the programmer who must take into account the repetition of the instruction.

BIT 1: $\overline{SL}$/PS

0 = Slave
1 = Pseudo Slave.

A pseudo-slave processor can address an external RAM using the system bus (AD0-AD7) as address lines for its own local bus. Consequently the system bus is no more available for exchanging data between the pseudo-slave processor and the bus master.

The pseudo-slave processor behaves differently from a slave processor since in case of exchange it must relinquish this bus to the master following an exchange protocol. (Reference to I/O)

BIT 2: $\overline{SB}$/CB

0 = Separate bus
1 = Concatenated bus.

The local bus can be used as two independent 8-bit buses (D0-D7), (D8-D15) or a single 16 bit-bus (D0-D15).

BIT 3: I/M

0 = Control pulses Read ($\overline{RD}$) and Write ($\overline{WR}$) are generated
1 = Control pulses data strobe ($\overline{DS}$) and Read/Write (R/$\overline{W}$) are generated.

The local bus supports the two main types of interchange signal:
- A slave processor, a data converter such as the MAFE, a 68000 peripheral, etc. requiring a data strobe and a read/write pulse.
- The standard bytewide RAM requiring a read and a write pulse.

BIT 4: $\overline{DTACK}$/BE6

0 = $\overline{DTACK}$ Indicates transfer acknowledge on the system bus to insure 68000 family compatibility.
1 = BE6 External test condition.

BIT 5: BA/BE5

0 = BA BUS available. Indicates to the master that the pseudo-slave is not using the system bus for generating addresses on local bus.
1 = BE5 External test condition.

BIT 6: MASK (TS68931 only)

0 = AMR is not masked. When an external halt is applied to the processor the AMR register does not change.
1 = AMR is masked. When an external halt is applied to the processor the AMR register changes to the following state: FAST EXCHANGE, PSEUDO-SLAVE, CONCATENATED BUS, $\overline{RD}$ and $\overline{WR}$ control pulses.

This bit can be modified by the programmer even while the $\overline{HALT}$ is asserted.

5.6 RESET

The reset signal has the following effects on the different blocks on the circuit :

SEQUENCER
PC, LC cleared to zero.
IR loaded with NOP instruction.

STATUS:
- REAL mode
- no saturation
- empty FIFO (EF = 1)
- memorised overflow (MOVF) = 0.

X or YRAM
- no circular addressing mode.

AMR
- Fast exchange
- Slave
- Concatenated bus
- $\overline{RD}$ and $\overline{WR}$
- BE6
- BE5.

RESET must be maintained for a minimum of 3 clock cycles (480 ns) to be effective.

5.7 HALT (TS68931 only)

The external halt signal will freeze the program counter and the loop counter. The instruction register can then be loaded from an external source. This signal is used for system development. If the MASK bit = 1 it will force the AMR into the following state: FAST EXCHANGE, SLAVE, SEPARATE BUS, RD and WR control pulses.

SECTION 6
INPUT/OUTPUT

6.1. DUAL-BUS INTERFACE

In order to permit a maximum versatility the processor interface provides two buses :
- the system Bus AD0-AD7
- the local Bus D0-D5.

This dual-bus interface allows the processor to be used in the following ways :
a) a microprocessor peripheral (fig. 6.A.)
b) a slave of another processor (fig. 6.B.)
c) a stand-alone unit connected to a peripheral or a data converter (fig. 6.C.)
d) a processor and its external memory (fig. 6.D.)
e) an intelligent peripheral connected to a general purpose microprocessor (fig. 6.E.)

These are some examples of the possibilities offered by the dual-bus interface. In addition very sophisticated multiprocessor machines can be built based on the principle of tree hierarchy (fig. 6.F.). In effect each processor becomes nested in the multiprocessor machine in the same way as subroutines are nested in a program tree.

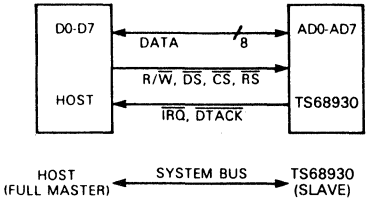


FIGURE 6.A. - HOST/TS68930

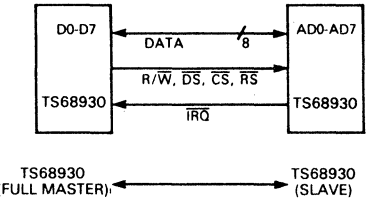


FIGURE 6.B. - TS68930/TS68930

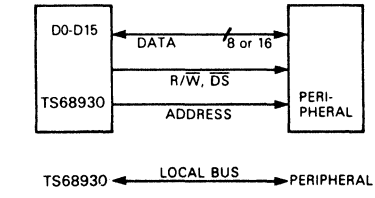


FIGURE 6.C. - TS68930/PERIPHERAL

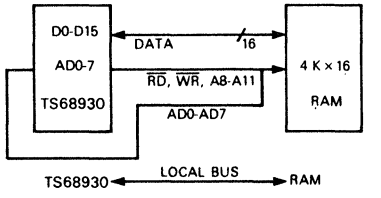


FIGURE 6.D. - TS68930/RAM

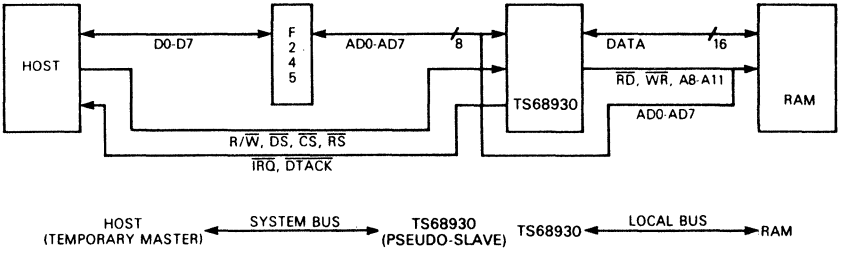


FIGURE 6.E. - TEMPORARY MASTER/PSEUDO-SLAVE

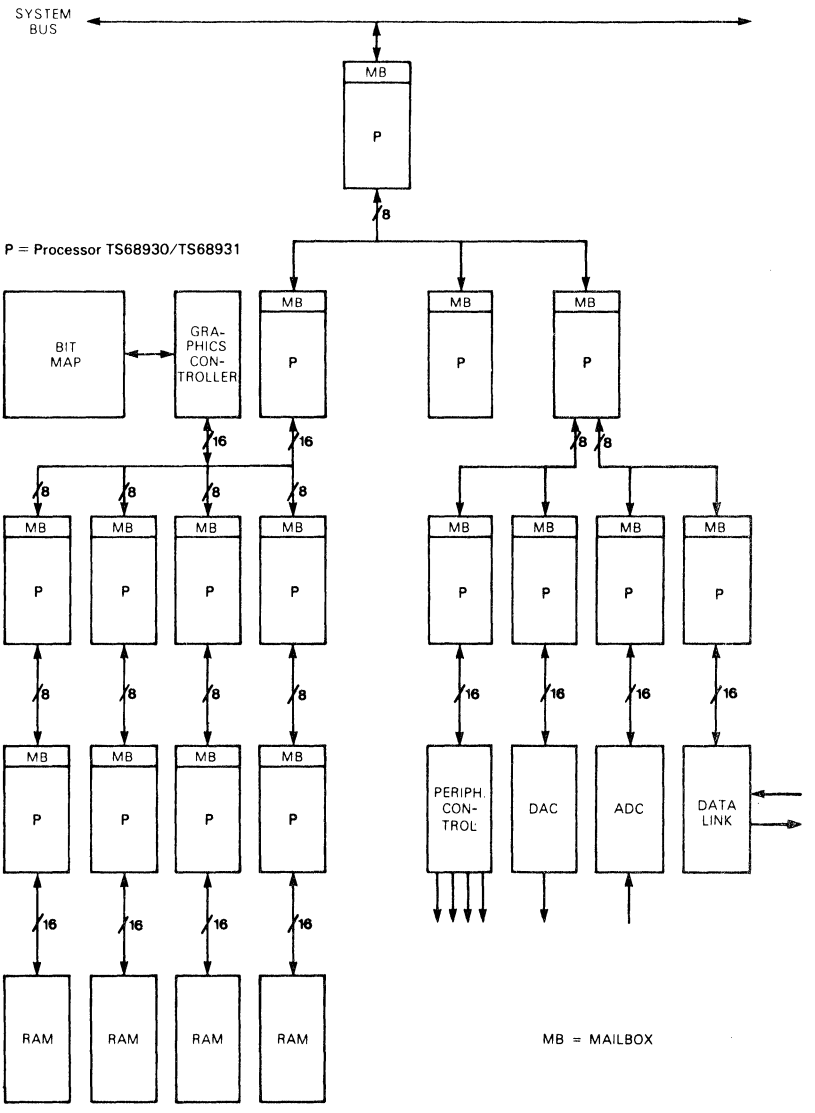


FIGURE 6.F. - MULTIPROCESSING MACHINE

6.2. MASTER/SLAVE

The processor is **a master on its local bus** and **a slave on its system bus.** There are times where the processor needs to access an external RAM and for that purpose will use the system bus to generate the addresses.

In this case this circuit prevents the master from using the bus freely and for that reason is called **a pseudo-slave.** Since the master can only gain access to the bus temporarily it is now defined as a **temporary master**.

It is the programmer who decides whether the processor should behave as a slave or a pseudo-slave.

This is done by programming the Access Mode Register.

That gives four different types of processor configurations:

PSI type	Definition
SLAVE (SL)	Its system bus is used to exchange data with a full master.
PSEUDO-SLAVE (PS)	Its system bus is also used to generate addresses for its local external memory.
FULL-MASTER (FM)	It has complete mastership of its local bus.
TEMPORARY-MASTER (TM)	Its local bus is shared with another processor which uses it to generate addresses.

These exchange type can be summarized to three possible connections:
1) Full master ↔ slave
2) Full master ↔ memories or peripherals
3) Temporary master ↔ pseudo-slave.

Connection 1 (with reference to fig. 6.A., 6.B.):

The data is exchanged through a mailbox and the exchange follows the mailbox protocol.

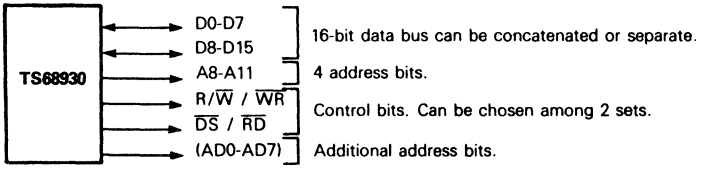
Connection 2 (example 6.C., 6.D., 6.E.):

The exchange is equivalent to reading and writing of data into locations or registers.

Connection 3 (example 6.E.):

The data is exchanged through a mailbox and the exchange follows the mailbox protocol.

6.3. LOCAL BUS PIN DESCRIPTION



$\overline{DS}$ = data strobe. Synchronizes the transfer.
R/$\overline{W}$ = indicates the direction of data.
$\overline{RD}$ = read clock pulse.
$\overline{WR}$ = write clock pulse.

The bus can take the form of two independent 8-bit buses or a single 16-bit bus.

There are four address bits (A8-A11) which are sufficient to address many slaves without requiring additionnal circuitry.

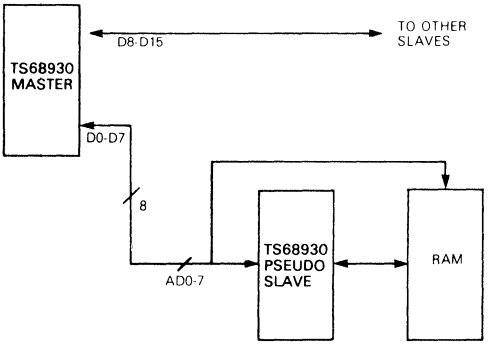
The address bus can be extended to 12 bits (AD0-AD7) to access an external memory.

If a peripheral is too slow to answer in one instruction cycle the processor can be programmed into a slow exchange mode. This mode is particularly useful for peripherals such as data converters, or the dedicated analogue interface circuit fabricated by THOMSON for modem applications. (The MAFE: Modem Analog Front-End).

SEPARATION OF LOCAL BUS

The processor offers the possibility of dividing the local bus D0-D15 into two independent 8-bit buses. This is used when a pseudo-slave monopolizes the bus to generate its own RAM addresses (fig. 6.3.) on D0-D7. By separating the bus, the processor can remain a full-master on D8-D15 even while being a temporary master on D0-D7, and it does not require the use of a bus transceiver on D0-D7.

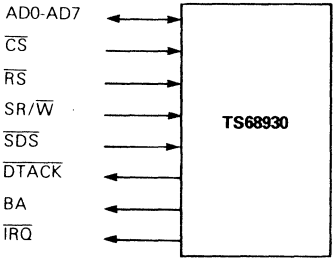
The selection between the 2 x 8-bit buses is made by the addresses A10-A11.



A11	A10	SELECTION
0	0	D0-D7
0	1	D8-D15
1	0	D8-D15
1	1	D8-D15

FIGURE 6.3. - SEPARATE LOCAL BUSES

6.4. SYSTEM BUS AND MAILBOX



AD0-AD7 = 8-bit data bus.
CS, RS = Mailbox control signal. Also used by master to gain access to bus.
SR/W = System Read/Write
SDS = System data strobe.
} Generated by external circuit (master)
IRQ = Handshake signal. Used by the master to gain access to mailbox (and bus).
DTACK = Data acknowledge. Compatibility with 68000 family.
BA = Bus available. The PSI is not currently using the system bus to generate addresses.

MAILBOX

The mailbox is comprised of two sets of registers : RIN and ROUT.

RIN (3 x 8-bit shift register).

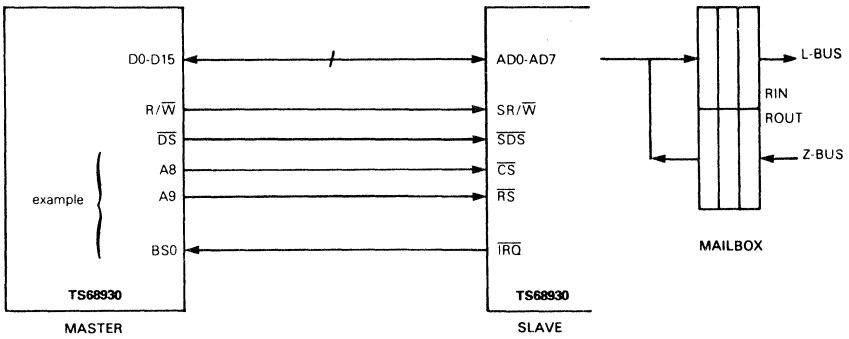
This register is read internally on the upper byte of L-BUS (L8-L15) and written externally from the system bus.

After each write operation (commanded by the external master) or slave read operation the data is shifted by 1.

ROUT (3 x 8-bit shift register).

This register is written internally with the upper byte of the Z-BUS (Z8-15) and read externally on the system bus by the external master. After each master read operation or slave write operation the data is shifted by 1.

6.5. MAILBOX PROTOCOL



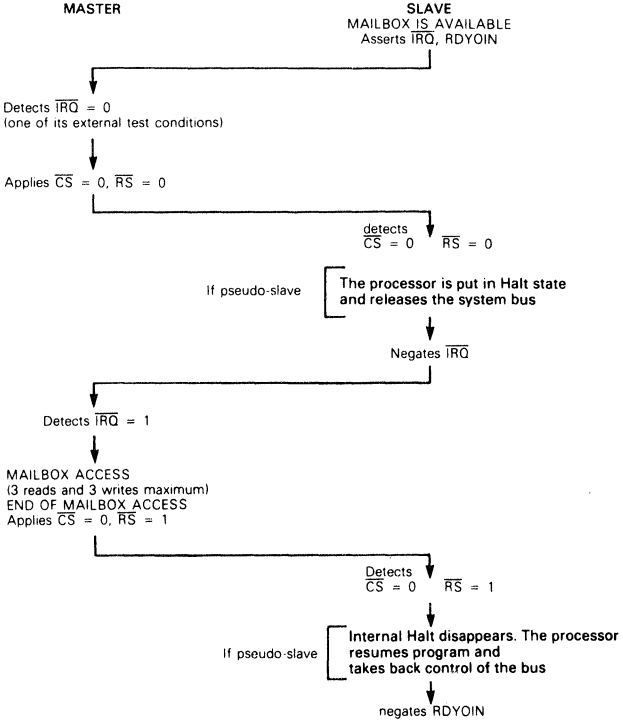
This protocol is hardwired on the slave side and programmed on the master side. The mailbox is included in the slave. The two slave address pins ($\overline{CS}$, $\overline{RS}$) are directly connected to two master address lines.

Therefore, the slave is seen as two external memory locations by the master which will address it by generating an external address directly or indirectly (pointer E0 or E1).

By addressing the location 00 the master echoes the $\overline{IRQ}$ to the slave and accesses the mailbox.

By addressing the location 01 the master releases the bus.

The complete protocol is explained below.



SIGNAL MEANING

RDYOIN

Internal flag indicating the property of the mailbox.
0 = Slave has access to the mailbox
1 = Master has access to the mailbox.

a) RDYOIN is set by the slave and reset by the master. That means that the slave gives the mailbox to the master when it finishes using it and vice-versa. In no case can the master or the slave request the mailbox, it can only wait for the other to give it back.

b) From the slave point of view, RDYOIN is a flag :
- tested by a branch instruction
- set by an initialization instruction.

$\overline{IRQ}$

Handshake signal used by the master to gain access to the mailbox;

a) **$\overline{IRQ}$ is asserted by the slave** to indicate the availability of the mailbox (at the same time as RDYOIN).

b) The master **(after testing $\overline{IRQ}$)** knows that it can access the mailbox but does not know if it has access to the bus (since it does not know if the slave is behaving as a pseudo-slave).
It requests the bus by generating the address $\overline{CS}$ = 0, $\overline{RS}$ = 0.

c) The slave internal I/O sequencer answers back by **negating $\overline{IRQ}$.** The master has now full control of the bus and the mailbox.
When the master has completed the exchange it generates the address CS = 0, $\overline{RS}$ = 1 and the slave internal I/O sequencer resets RDYOIN.

HALT (internal)

The internal halt has the following effects on the circuit :
- the program is stopped at the end of the current instruction; the program and loop counters are frozen
- a NOP is generated on the instruction bus
- no more addresses are generated on the system bus.

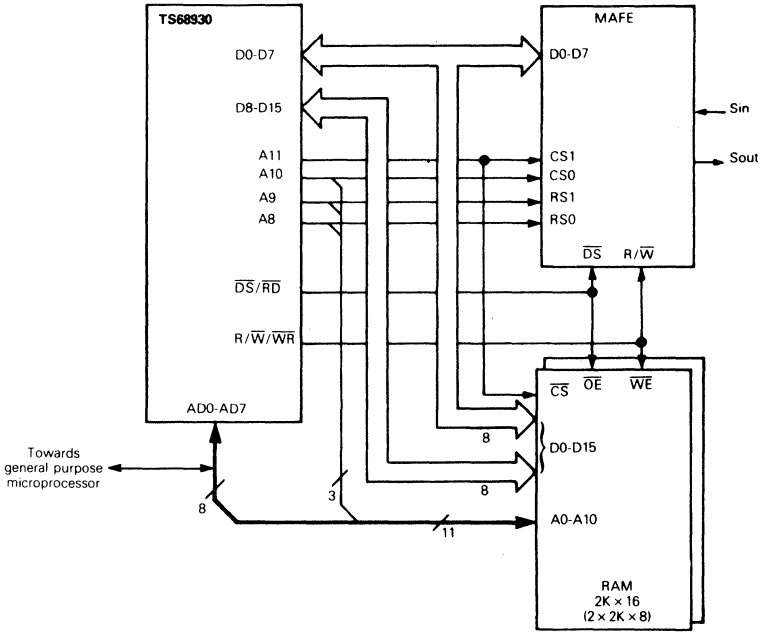
6.6. INSTRUCTION BUS (TS68931 only)

For the TS68931, CROM (512 x 16-bit) and IROM (64k x 32-bit) are external. They are read using the I-BUS, on which are multiplexed:
— the 16-bit instruction ROM address
— the 9-bit coefficient ROM address + 1 Output Enable bit (ENCROM)
— the 32-bit instruction code.

In order to synchronize the exchanges, an additional signal is generated: INCYCLE.
It is the internal instruction clock.

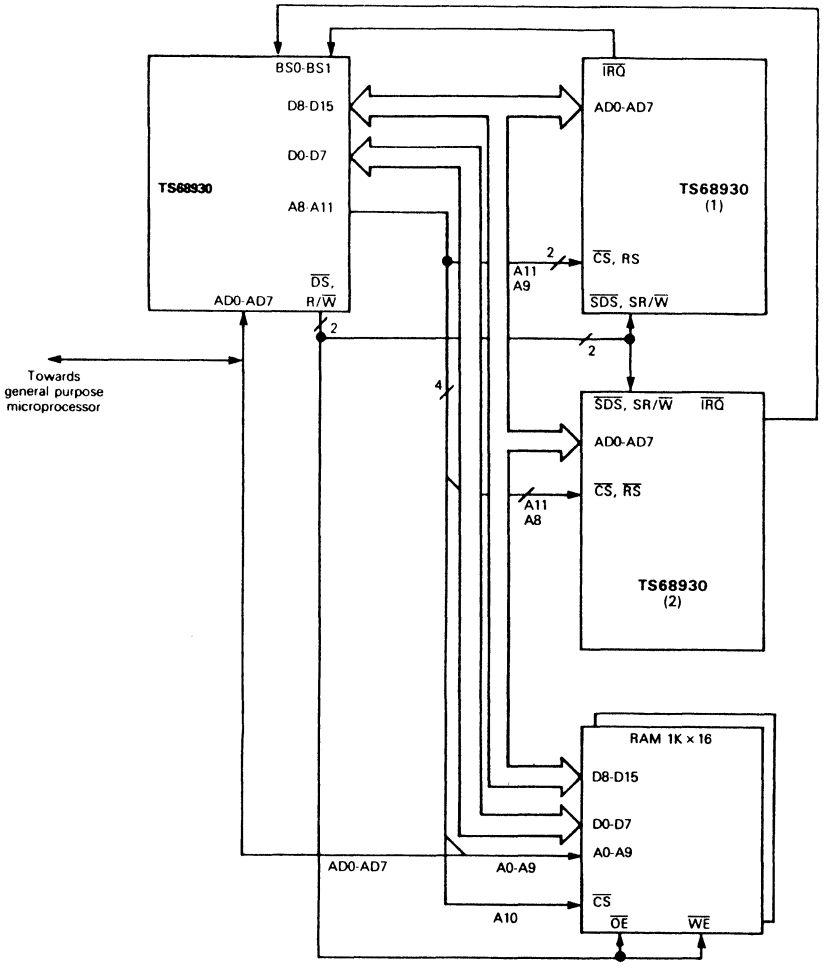
Data from CROM are read on the local bus.

6.7. APPLICATION EXAMPLES



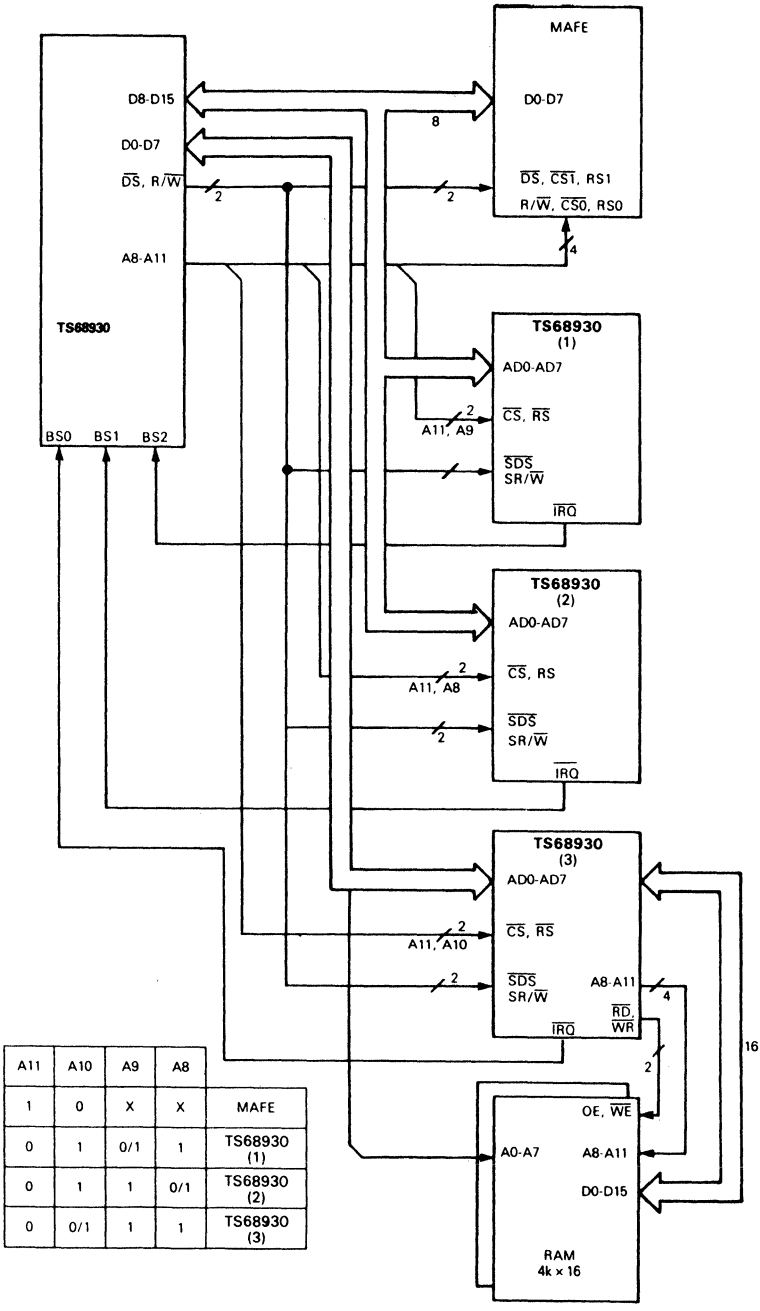
A11	A10	A9	A8	
1	0	X	X	MAFE
0	X	X	X	RAM

FIGURE 6.7.A. - CONFIGURATION EXAMPLE: TS68930 + RAM + MAFE



A11	A10	A9	A8	
0	1	0/1	1	TS68930 (1)
0	1	1	0/1	TS68930 (2)
1	0	X	X	RAM

FIGURE 6.7.B. - CONFIGURATION EXAMPLE: 3 TS68930 + RAM



A11	A10	A9	A8	
1	0	X	X	MAFE
0	1	0/1	1	TS68930 (1)
0	1	1	0/1	TS68930 (2)
0	0/1	1	1	TS68930 (3)

FIGURE 6.7.C. - CONFIGURATION EXAMPLE: 4 TS68930 + MAFE + RAM

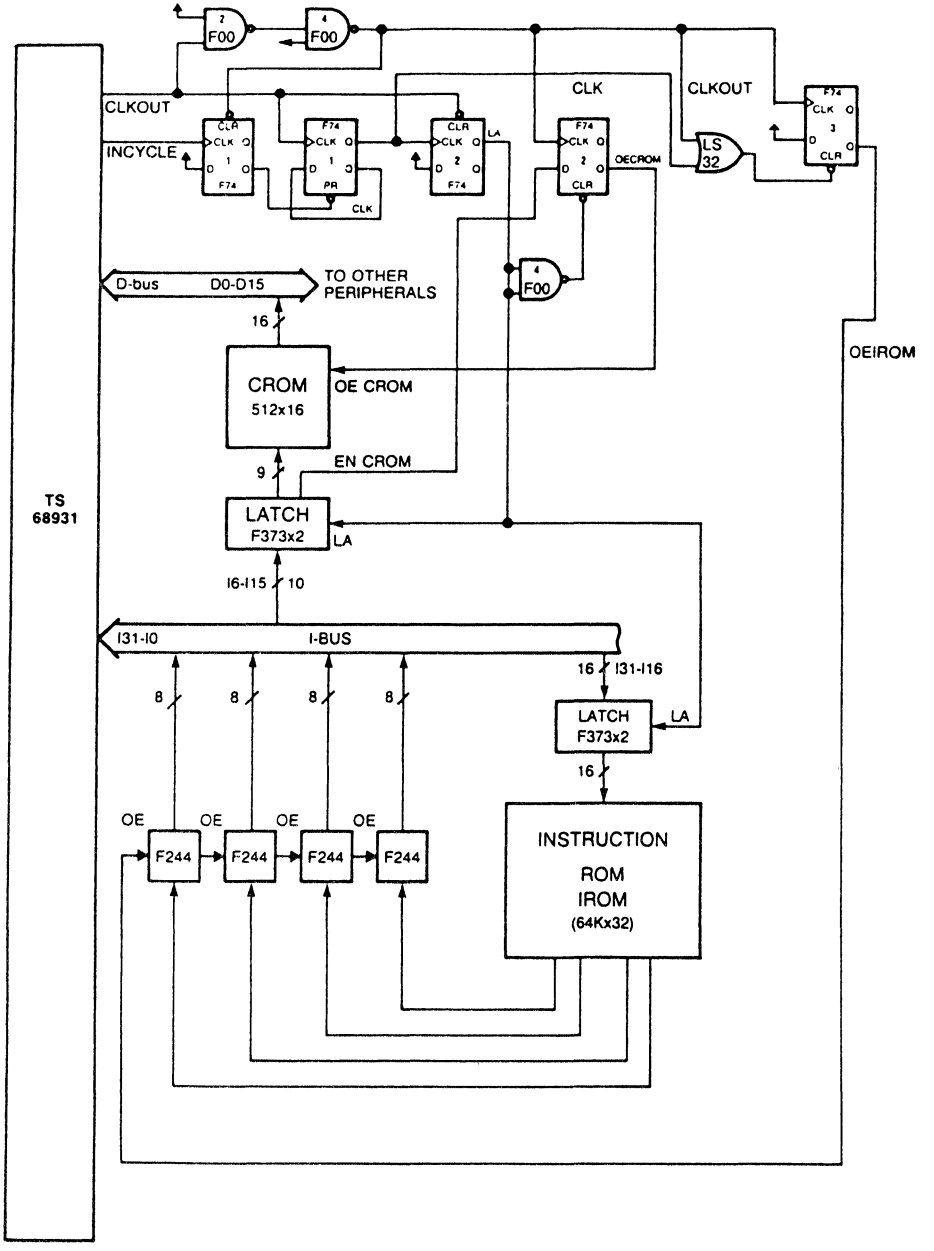


FIGURE 6.7.D. - I-BUS INTERFACE (TS68931)

SECTION 7
INSTRUCTION SET

Type	Mnemonic	Operation	Number of cycles	
			REAL	CPLX DBPR
Calculation, instruction with indirect addresing	OPIN	This instruction refers to operands indirectly addressed	1	2
Calculation, instruction with direct addressing	OPDI	The operand sourcing the L-BUS is directly addressed	1	2
Calculation, instruction with immediate operand	OPIM	An immediate operand is read on R-RUS	1	2
General shift Instruction	ASR ASL LSR ROR	The operand sourcing the L-BUS can be shifted/rotated by 0 → 15 bits	1	2
Immediate branch Instruction	BRI	Conditional / unconditional branch to direct address	2	2
Computed branch Instruction	BRC	Conditional / unconditional branch to computed address	2	2
Data transfer Instruction	SVR	This instruction is used to save register contents in external or internal RAM	1	2
Initialization and control instruction	INI	Pointers, access mode register, loop counter, mode initialization	1	2

INSTRUCTION SET LANGUAGE DEFINITIONS

LDT	Load L-BUS source into transfer register T
R SRC	R-BUS source
L SRC	L-BUS source
SL	ALU input selection - left side
SR	ALU input selection - right side
ALU DST	ALU output destination
ALUCODE	ALU codes
LDM	Load L-BUS source into multiplier input M
LDN	Load R-BUS source into multiplier input N
Z SRC	Z-BUS source
Z DST	Z-BUS destination
ZT	Load Z-BUS into transfer register T
ACE	Post incrementation: pointers CROM or ERAM
AY	Post incrementation: pointers YRAM
AX	Post incrementation: pointers XRAM
BRA	Branch address source
FT	False / True condition
SVPC	Save program counter
JDST	Destination register for J constant
KDST	Destination register for K constant
MODE	Operating mode
SAT	Saturation flag
ADOF	Even / odd flag
J7	YRAM circular addressing mode flag
J constant	8-bit constant used to initialize registers
K7	XRAM circular addressing mode flag
K constant	12-bit constant used to initialize registers.

7.1. OPERATING CODE FORMATS

Bit	Field	Operations and codes
0 1	OP CODE	00
2	LDT	0-NO LOAD, 1-LBUS → T
3 4	R SRC	00 01 10 11 [X0] [E0] [Y0] [Y1]
5 6 7	L SRC	000 001 010 011 100 101 110 111 [X0] [X1] [Y0] RIN T [E1] [C0] [C1]
8	SL	0-LBUS / 1-P
9	SR	0-RBUS / 1-A/B (REFER TO ALU DST)
10 11 12 13 14	ALU CODE	CF. SPECIAL TABLE
15 16	ALU DST	00 01 10 11 D F A B
17 18 19	Z SRC	000 001 010 011 100 101 110 111 D F A B T CCR — —
20	LDM	0-NO LOAD / 1-LBUS → M
21	LDN	0-NO LOAD / 1-RBUS → N
22 23	ACE	00 01 10 11 +0 +1 — −1
24 25	AY	00 01 10 11 +0 +1 — −1
26 27	AX	00 01 10 11 +0 +1 — −1
28 29 30	Z DST	000 001 010 011 100 101 110 111 NONE ROUT [Y0] [Y1] [E0] [E1] [X0] [X1]
31	ZT	0-NO LOAD / 1-ZBUS → T

FIGURE 7.1. - OPIN: CALCULATION INSTRUCTION WITH INDIRECT ADDRESSING

Bit	Field	Operations and codes
0 1 2	OP CODE	010
3 4	R SRC	00 01 10 11 [X0] [E0] [Y0] [Y1]
5 6 7	L SRC	000 001 010 011 100 101 110 111 X — Y RIN T E — C
8	Z SRC	0-D / 1-F
9	SR	0-RBUS / 1-A
10 11 12 13 14	ALU CODE	CF. SPECIAL TABLE
15	ALU DST	0-F / 1-A
16 17 18 19 20 21 22 23 24 25 26 27	LBUS DIRECT ADDRESS	MSB LSB
28 29 30 31	Z DST	0000 0010 0100 0110 1000 1010 1100 1110 NONE ROUT [Y0] [Y1] [E0] [E1] [X0] LCR 0001 0011 0101 0111 1001 1011 1101 1111 X0 X1 Y0 Y1 E0 E1 C0 C1

FIGURE 7.2. - OPDI: CALCULATION INSTRUCTION WITH DIRECT ADDRESSING

OPERATING CODE FORMATS (Continued)

Bit	Field	Operations and codes
0 1 2 3 4	**OP CODE**	**01110**
5 6 7	**L SRC**	000 [X0] 001 [X1] 010 [Y0] 011 **RIN** 100 **T** 101 [E1] 110 [C0] 111 [C1]
8	**SL**	0-LBUS / 1-P
9	**SR**	0-RBUS / 1-A
10 11 12 13 14	**ALU CODE**	**CF. SPECIAL TABLE**
15	**ALU DST**	0-F / 1-A
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	**IMMEDIATE VALUE**	**MSB** **LSB**

FIGURE 7.3. - OPIM: CALCULATION INSTRUCTION WITH IMMEDIATE OPERAND

Bit	Field	Operations and codes
0 1 2 3 4	**OP CODE**	**01111**
5 6 7	**L SRC**	000 **X** 001 **—** 010 **Y** 011 **RIN** 100 **T** 101 **E** 110 **—** 111 **C**
8	**SL**	0-LBUS / 1-P
9 10	**ALU CODE**	00 **ASR** 01 **LSL** 10 **LSR** 11 **ROR** } **NOTE:** When LSR, ASR, ROR shift value is complemented to 2
11 12 13 14	**SHIFT VALUE**	0000 **0** 0001 **1** .... 1111 **15** }
15	**ALU DST**	0-F / 1-A
16 17 18 19 20 21 22 23 24 25 26 27	**LBUS DIRECT ADDRESS**	**MSB** **LSB**
28 29 30 31		

FIGURE 7.4. - ASR, LSL, LSR, ROR: SHIFT INSTRUCTIONS

OPERATING CODE FORMATS (Continued)

Bit	Field	Operations and codes
0 1 2	OP CODE	100
3	BRA	0-IR, 1-RAR
4	FT	0-FALSE, 1-TRUE
5 6 7 8	COND	REFER TO SPECIAL TABLE
9	SVPC	0-NO SVPC, 1-PC → RAR
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	BRANCH ADDRESS	MSB LSB
26 27	AX	00 01 10 11 +0 +1 – –1
28 29 30	Z DST	000 001 010 011 100 101 110 111 NONE – [Y0] [Y1] – – [X0] [X1]
31	ZT	0-NO LOAD, 1-ZBUS → T

FIGURE 7.5. - BRI: IMMEDIATE BRANCH INSTRUCTION

Bit	Field	Operations and codes
0 1 2 3	OP CODE	1010
4	FT	0-FALSE, 1-TRUE
5 6 7 8	COND	REFER TO SPECIAL TABLE
9	SVPC	0-NO SVPC, 1-PC → RAR
10 11 12 13 14 15		
16 17 18 19	BRANCH SOURCE	000 001 010 011 100 101 110 111 – F A B T – – –
20 21 22 23 24 25		
26 27	AX	00 01 10 11 +0 +1 – –1
28 29 30	Z DST	000 001 010 011 100 101 110 111 NONE – [Y0] [Y1] – – [X0] [X1]
31	ZT	0-NO LOAD, 1-ZBUS → T

FIGURE 7.6. - BRC: COMPUTED BRANCH INSTRUCTION

OPERATING CODE FORMATS (Continued)

Bit	Field	Operations and codes
0 1 2 3 4 5	OP CODE	011000
6 7 8 9	Z SRC	0000 0001 0010 0011 0100 0101 0110 0111 X0 X1 Y0 Y1 E0 E1 C0 C1 1000 1001 1010 1011 1100 1101 1110 1111 AMR LC A F D STA — —
10 11 12 13 14 15		
16 17 18 19 20 21 22 23 24 25 26 27	ZBUS DIRECT ADDRESS	MSB LSB
28 29 30	Z DST	000 001 010 011 100 101 110 111 NONE ROUT Y — E — X —
31	ZT	0-NO LOAD, 1-ZBUS → T

FIGURE 7.7. - SVR: DATA TRANSFER INSTRUCTION

Bit	Field	Operations and codes
0 1	OP CODE	11
2 3 4	J DST	000 001 010 011 100 101 110 111 AMR LCD Y0 Y1 — RDYOIN EF NONE
5 6 7	K DST	000 001 010 011 100 101 110 111 X0 X1 LCI-LCR NONE E0 E1 C0 C1
8 9	MODE	00 01 10 11 — REAL DBPR CPLX
10	SAT	0 NO SATURATION MODE 1 SATURATION MODE
11	ADOF	0 NO INVERSION 1 INVERSION LSB ADDRESS X/Y RAM
12 13 14 15 16 17 18 19	J7 J CONSTANT J0	J7 0 YRAM NORMAL MODE 1 YRAM CIRCULAR ADDRESSING MODE
20 21 22 23 24 25 26 27 28 29 30 31	K11 K CONSTANT K0	K7 0 XRAM NORMAL MODE 1 XRAM CIRCULAR ADDRESSING MODE

FIGURE 7.8. - INI: INITIALIZATION AND CONTROL INSTRUCTION

7.2. ALU CODES

MNEMO-	Function	SR	SI	CR	CI	Z	OVF	MOVF	AOVF	CODE
ADD	A + B	*	*	*	*	*	*	*	*	00010
ADDC	A + B + CARRY	*	*	*	*	*	*	*	*	00011
ADDS	B + A/16	*	*	*	*	*	*	*	*	00001
ADDX	B + A* (COMPLEX CONJUGATE)	*	*	*	*	*	*	*	*	01010
AND	A . B	*	*	0	0	*	0	–	*	01110
ASL	CARRY ← [ ] ← 0	*	*	*	*	*	*	–	*	01011
ASR	[ ] → CARRY	*	*	*	*	*	0	–	*	01111
CLR	CLEAR	0	0	0	0	1	0	–	0	10011
COM	COMPLEMENT A	*	*	0	0	*	0	–	*	10110
COM	COMPLEMENT B	*	*	0	0	*	0	–	*	11000
LCCR	LBUS → CCR	*	*	*	*	*	*	*	*	01001
LSL	CARRY ← [ ] ← 0	*	*	*	*	*	0	–	*	11011
LSLB	LSL BYTE	*	*	*	*	*	0	–	*	11001
LSR	0 → [ ] → CARRY	*	*	*	*	*	0	–	*	00111
LSRB	LSR BYTE	*	*	*	*	*	0	–	*	11010
NOP		–	–	–	–	–	–	–	–	00000
OR	A ∧ B	*	*	0	0	*	0	–	*	01101
RCE	EXECUTE RC	*	*	*	*	*	*	*	*	10001
RCER	EXECUTE RC / LOAD NEW CODE	*	*	*	*	*	*	*	*	10000
RCR	LOAD RC									10010
ROR	[ ] → CARRY	*	*	*	*	*	0	–	*	10111
SBC	A + $\overline{B}$ + CARRY	*	*	*	*	*	*	*	*	00101
SBCR	$\overline{A}$ + B + CARRY	*	*	*	*	*	*	*	*	01000
SET		*	*	0	0	0	0	–	0	11100
SUB	A + $\overline{B}$ + 1	*	*	*	*	*	*	*	*	00100
SUBR	$\overline{A}$ + B + 1	*	*	*	*	*	*	*	*	00110
TRA	TRANSFER A	*	*	0	0	*	0	–	*	10100
TRA	TRANSFER B	*	*	0	0	*	0	–	*	10101
XOR	A ⊕ B	*	*	0	0	*	0	–	*	01100

* Affected bit.

Notes :

1) A/B refer to ALU inputs (RESP. LSIDE/RSIDE) not to accumulators A/B

2) In ASL the Carry bit is equivalent to exclusive - or of bit 14 and 15.

7.3. TEST CONDITIONS

TRUE CONDITION	FALSE CONDITION	CODE
BE3	NO BE3	0100
BE4	NO BE4	0010
BE5	NO BE5	0011
BE6	NO BE6	0001
BRANCH ALWAYS	BRANCH NEVER	0000
BS0	NO BS0	1100
BS1	NO BS1	1101
BS2	NO BS2	1110
CI	NO CI	1010
CR	NO CR	0110
MOVF	NO MOVF	1011
OVF	NO OVF	0111
RDYOIN	NO RDYOIN	1111
SI	NO SI	1001
SR	NO SR	0101
Z	NO Z	1000

SECTION 8
PERFORMANCE EVALUATION

	TIME (μs)
TRANSVERSAL FILTER (N COEFFICIENTS) (1)	
REAL COMPLEX ADAPTIVE REAL ADAPTIVE CMPLX	0.160 x N 0.320 x N 0.320 x N 0.640 x N
BIQUAD FILTER - 4 COEFF	0.960
LATTICE FILTER - 10 STAGE (1)	6.4
AUTOCORRELATION - 10 th ORDER (2) (240 samples) (32-bit result)	 8 μs/sample 1.8 ms total
FFT (RADIX 2 - DIF - algorithm) (2) (3)	
64 - POINT COMPLEX 128 - POINT REAL 256 - POINT COMPLEX	 265 270 2000
COSINE CALCULATION	2.4

Notes :
(1) Excluding initialization, context switching, pipeline.
(2) Using external RAM.
(3) Including loading/unloading, scaling, bit reserve.

SECTION 9
ELECTRICAL SPECIFICATIONS

9.1. MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply voltage	V_{CC}*	−0.3 to 7.0	V
Input voltage	V_{in}*	−0.3 to 7.0	V
Operating temperature range	T_A	0 to 70	°C
Storage temperature range	T_{stg}	−55 to 150	°C
Max. power dissipation	P_{Dmax}	3	W

* With respect to V_{SS}

Stresses above those hereby listed may cause permanent damage to the device. The ratings are stress ones only and functional operation of the device at these or any conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. Standard MOS circuits handling procedure should be used to avoid possible damage to the device.

9.2. DC ELECTRICAL CHARACTERISTICS

V_{CC} = 5.0 V ± 5 %, V_{SS} = 0, T_A = 0 to + 70°C (Unless otherwise specified)

Characteristic	Symbol	Min	Typ	Max	Unit
Supply voltage	V_{CC}	4.75	5	5.25	V
Input low voltage	V_{IL}	−0.3	–	0.8	V
Input high voltage	V_{IH}	2.4	–	V_{CC}	V
Input leakage current	I_{in}	–	–	10	μA
Output high voltage (I_{load} = −300 μA)	V_{OH}	2.7	–	–	V
Output low voltage (I_{load} = 3.2 mA)	V_{OL}	–	–	0.5	V
Power dissipation	P_D	–	1.5	–	W
Input capacitance	C_{in}	–	10	–	pF
Three state (off state) input current	I_{TSI}	–	–	10	μA

9.3. AC ELECTRICAL SPECIFICATIONS - CLOCK AND CONTROL PINS TIMING

(V_{CC} = 5.0 V ± 5 %, T_A = 0° to + 70°C; see figure 9.1.)

OUTPUT LOAD = 50 pF + DC characteristics I load

REFERENCE LEVELS: V_{IL}: 0.8 V V_{IH}: 2.4 V
V_{OL}: 0.8 V V_{OH}: 2.4 V

tr, tf ≤ 5 ns for input signals

Characteristic	Symbol	Min	Typ	Max	Unit
External clock cycle time	tcex	40		160	ns
External clock fall time	tfex			5	ns
External clock rise time	trex			5	ns
EXTAL to CLKOUT high delay	tcoh		25		ns
EXTAL to CLKOUT low delay	tcol		25		ns
CLKOUT rise time	tcor			10	ns
CLKOUT fall time	tcof			10	ns
CLKOUT to $\overline{DS}$, $\overline{RD}$, $\overline{WR}$ low	tdsl		5		ns
CLKOUT to DS, RD, WR high	tdsh		5		ns
Control inputs set-up time (BS0...BS2, BE3...BE6, Reset, halt)	tsc	20			ns
Control inputs hold time (BS0...BS2, BE3...BE6, $\overline{Reset}$, $\overline{halt}$)	thc	10			ns
CLKOUT to control output low ($\overline{IRQ}$, BA)	tdlc			50	ns
CLKOUT to control output high (BA)	tdhc			50	ns

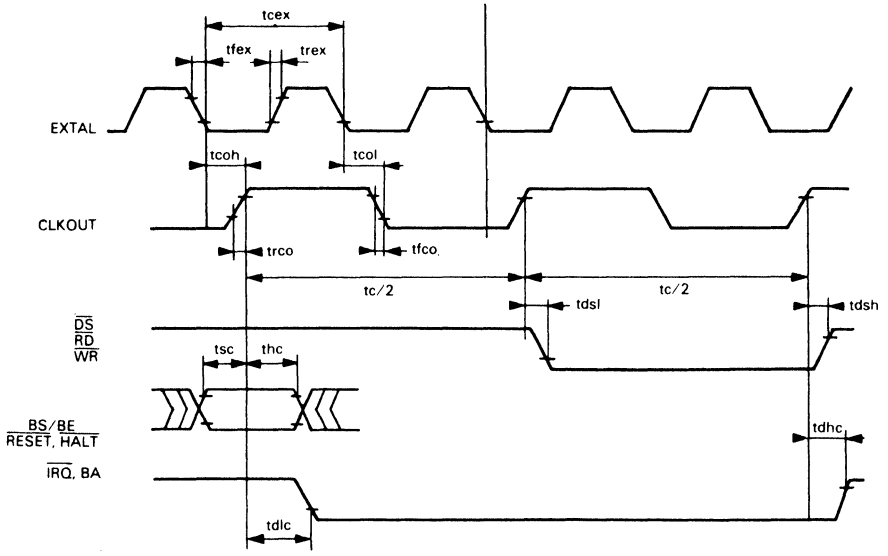
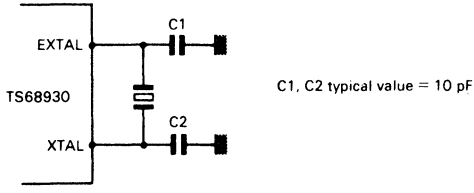


FIGURE 9.1. - CLOCK AND CONTROL PINS TIMING

INTERNAL CLOCK OPTION

A crystal oscillator can be connected across XTAL and EXTAL. The frequency of CLKOUT: tc/2 is half the crystal fundamental frequency.



9.4. AC ELECTRICAL SPECIFICATIONS - LOCAL BUS TIMING

(V_{CC} = 5.0 V ± 5 %, T_A = 0° to + 70°C; see figure 9.2.)

Characteristic	Symbol	Min.	Max.	Unit
$\overline{RD}$, $\overline{WR}$, $\overline{AS}$ pulse width	t_{PW}	1/2 tc - 15	1/2 tc	ns
address hold time	t_{AH}	10	–	ns
data set-up time, write cycle	t_{DSW}	25	–	ns
data hold time, write cycle	t_{DHW}	10	–	ns
data set-up time, read cycle	t_{DSR}	20	–	ns
data hold time, read cycle	t_{DHR}	5	–	ns
address valid to $\overline{WR}$, $\overline{AS}$, $\overline{RD}$ low	t_{ARW}	1/2 tc - 40	–	ns

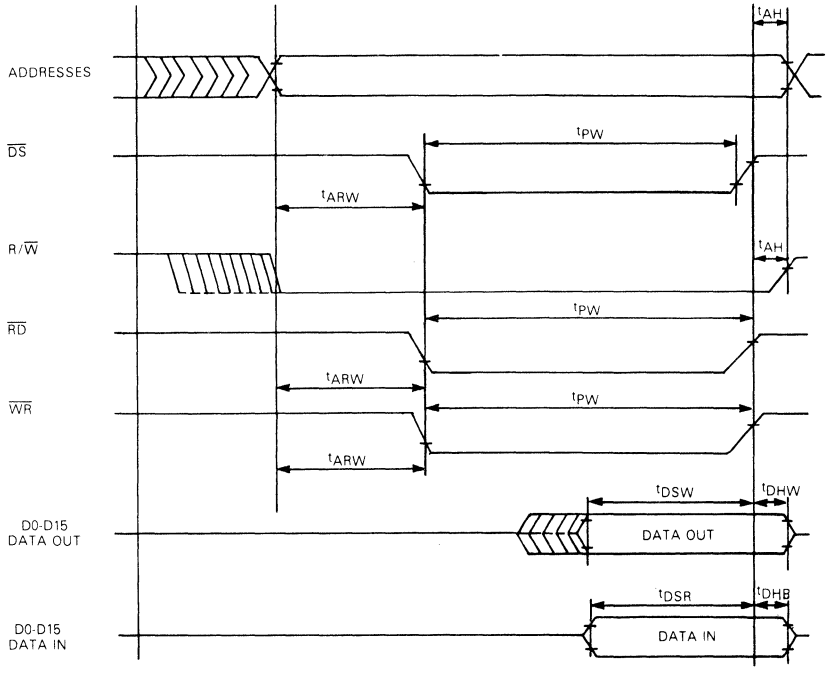


FIGURE 9.2. - LOCAL BUS TIMING DIAGRAM

9.5. AC ELECTRICAL SPECIFICATIONS - SYSTEM BUS TIMING

(V_{CC} = 5.0 V ± 5 %, T_A = 0° to + 70°C; see figure 9.3.)

Characteristic	Symbol	Min.	Max.	Unit
$\overline{SDS}$ pulse width	t_{SPW}	60	–	ns
SR/$\overline{W}$, $\overline{CS}$, $\overline{RS}$ set-up time	t_{SAW}	20	–	ns
SR/$\overline{W}$, $\overline{CS}$, $\overline{RS}$ hold after $\overline{SDS}$ high	t_{SAH}	5	–	ns
data set-up time, read cycle	t_{SDSR}	20	–	ns
data hold time, read cycle	t_{SDHR}	5	–	ns
data set-up time, write cycle	t_{SDSW}	–	35	ns
data hold time, write cycle	t_{SDHW}	10	50	ns
$\overline{SDS}$ low to $\overline{DTACK}$ low	t_{DSLDT}	–	50	ns
$\overline{SDS}$ high to $\overline{DTACK}$ high*	t_{DSHDT}	–	50	ns
$\overline{SDS}$ high to $\overline{IRQ}$ high	t_{DSHIR}		50	ns

* DTACK is an open drain output test load include R_L = 820 Ω at V_{CC}

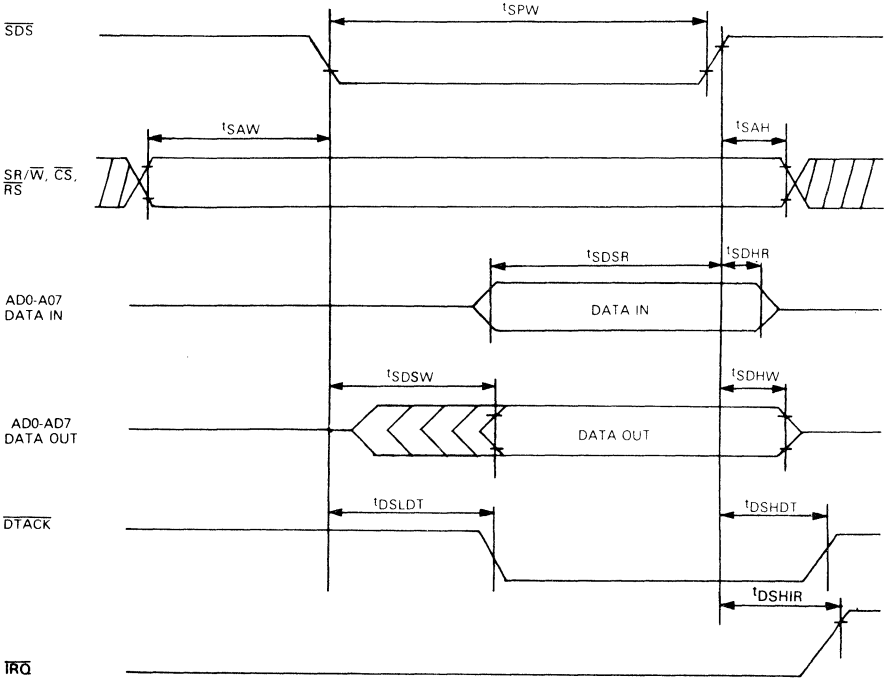


FIGURE 9.3. - SYSTEM BUS TIMING DIAGRAM

9.6. - AC ELECTRICAL SPECIFICATIONS - INSTRUCTION BUS TIMING

(V_{CC} = 5.0 V ± 5 %, T_A = 0° to + 70°C; see figure 9.4.)

Characteristic	Symbol	Min	Max	Unit
CLKOUT high to INCYCLE high	t_{INCH}	5	15	ns
CLKOUT low to INCYCLE low	t_{INCL}	5	15	ns
CLKOUT high to address valid	t_{IASW}		40	ns
I-BUS address hold	t_{IAHW}	20	40	ns
Instruction valid	t_{IISR}	20		ns
Instruction hold	t_{IIHR}	10		ns
CROM data set-up time	t_{DSR}	tc/2 - 40		ns
CROM data hold time	t_{DHR}	5		ns

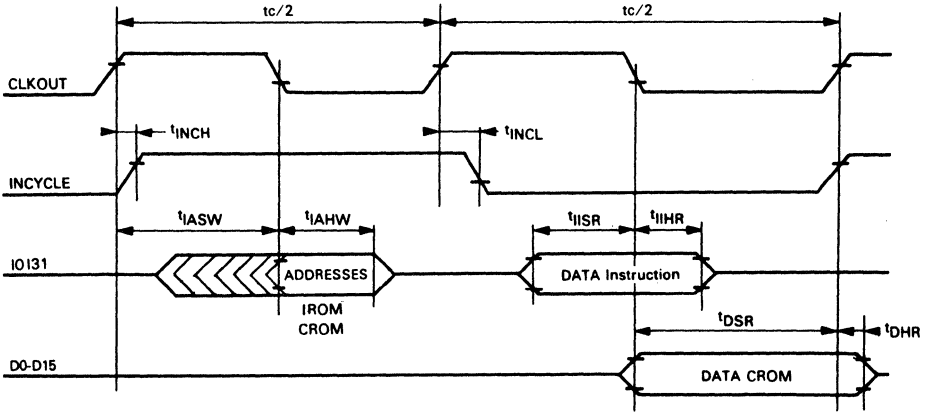
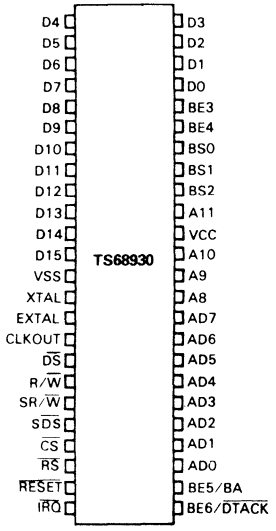


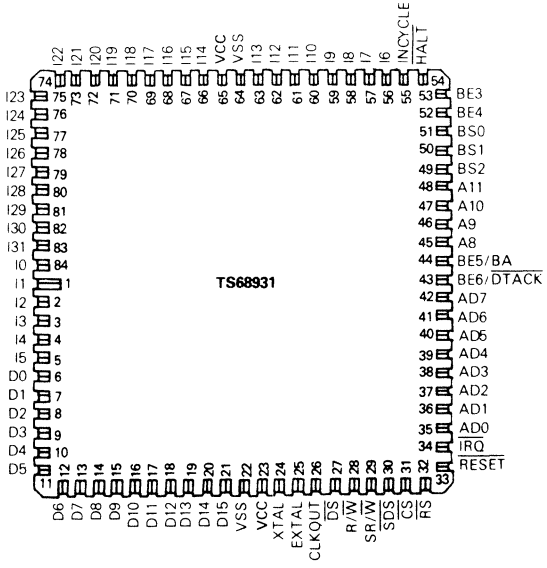
FIGURE 9.4. - I-BUS TIMING DIAGRAM

SECTION 10
PIN ASSIGNMENTS AND MECHANICAL DATA

10.1. - PIN ASSIGNMENTS



48-Pin Dual-in-Line Package



84-Terminal Chip Carrier (LCCC)

TS68931 84-Pin Grid Array (see table 10-1 Pin Assignments)

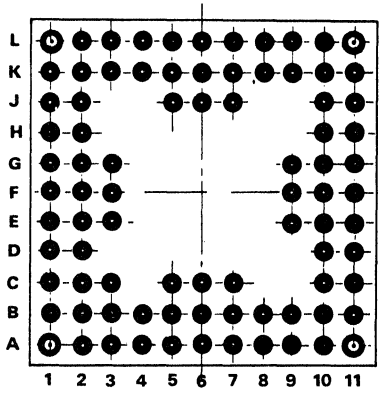
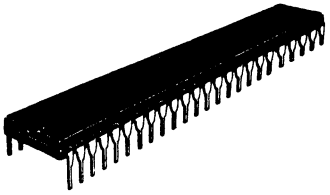


TABLE 10-1 PIN ASSIGNMENTS

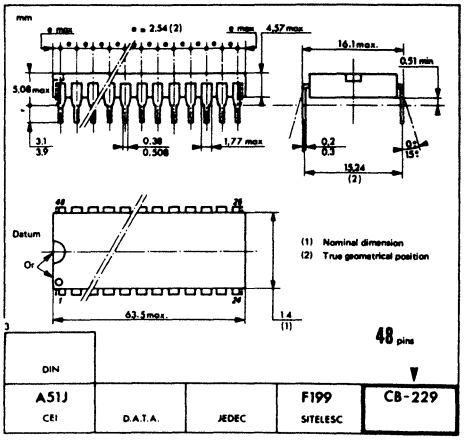
A1	I22	B11	B50	F9	BE6/$\overline{\text{DTACK}}$	K2	D6
A2	I20	C1	I26	F10	AD4	K3	D7
A3	I19	C2	I24	F11	BE5/BA	K4	D10
A4	I17	C5	I16	G1	I3	K5	D13
A5	I14	C6	VSS	G2	I4	K6	D15
A6	VCC	C7	I13	G3	I2	K7	EXTAL
A7	I12	C10	BE4	G9	AD7	K8	R/$\overline{\text{W}}$
A8	I9	C11	BS1	G10	AD5	K9	$\overline{\text{CS}}$
A9	I7	D1	I28	G11	AD6	K10	$\overline{\text{RESET}}$
A10	I6	D2	I27	H1	D0	K11	AD0
A11	BE3	D10	BS2	H2	D1	L1	D5
B1	I25	D11	A11	H10	AD2	L2	D8
B2	I23	E1	I31	H11	AD3	L3	D9
B3	I21	E2	I30	J1	D2	L4	D11
B4	I18	E3	I29	J2	D4	L5	D14
B5	I15	E9	A10	J5	D12	L6	CLKOUT
B6	I10	E10	A9	J6	VSS	L7	XTAL
B7	I11	E11	A8	J7	VCC	L8	$\overline{\text{DS}}$
B8	I8	F1	I5	J10	$\overline{\text{IRQ}}$	L9	SR/$\overline{\text{W}}$
B9	INCYCLE	F2	I0	J11	AD1	L10	SDS
B10	$\overline{\text{HALT}}$	F3	I1	K1	D3	L11	$\overline{\text{RS}}$

10.2. - PACKAGE DIMENSIONS

CB-229



TS68930
P SUFFIX
PLASTIC PACKAGE



SECTION 11
ORDERING INFORMATION

Package Type	Temperature Range	Part Number
Plastic DIL P Suffix	0°C to + 70°C −40°C to + 85°C	TS68930CP TS68930VP
Ceramic DIL C Suffix	0°C to + 70°C −40°C to + 85°C	TS68930CC TS68930VC
LCCC E Suffix	0°C to + 70°C −40°C to + 85°C	TS68931CE TS68931VE
PGA R Suffix	0°C to + 70°C	TS68931CR

As the TS68930 is a programmable circuit, a special ordering procedure has to be used. In order to get information about this procedure as well as the customer ordering sheet, please contact our sales representatives.

THOMSON COMPONENTS MOSTEK

TS68950

MODEM TRANSMIT ANALOG INTERFACE

COMMUNICATIONS PRODUCTS

DATA SHEET

The TS68950 is a transmit (Tx) analog front-end circuit designed to implement high speed voice-grade modems up to 19200 bps according to the CCITT V.22, V.26, V.27, V.29, V.32 and V.33 recommendations or the BELL 212A, 208 and 209 standards. This circuit is particularly suited to work with the TS68951 receive (Rx) analog front-end circuit, the TS68952 clock generator and the TS68930/31 digital signal processors (DSPs).

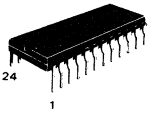
Main features

- Two-channel digital to analog converter (DAC) for Tx and echo-cancelling signals.
- 6th-order low-pass filter (switched-capacitor filter with output continuous-time smoothing cell).
- Programmable attenuation over a 22 dB range with 2 dB steps.
- Direct interface with MPU standard 8 bit bus.

CMOS

MODEM TRANSMIT ANALOG INTERFACE

CASE CB-68

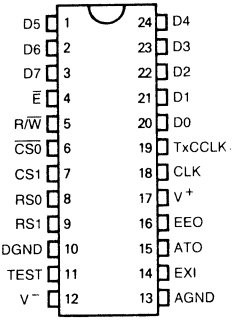


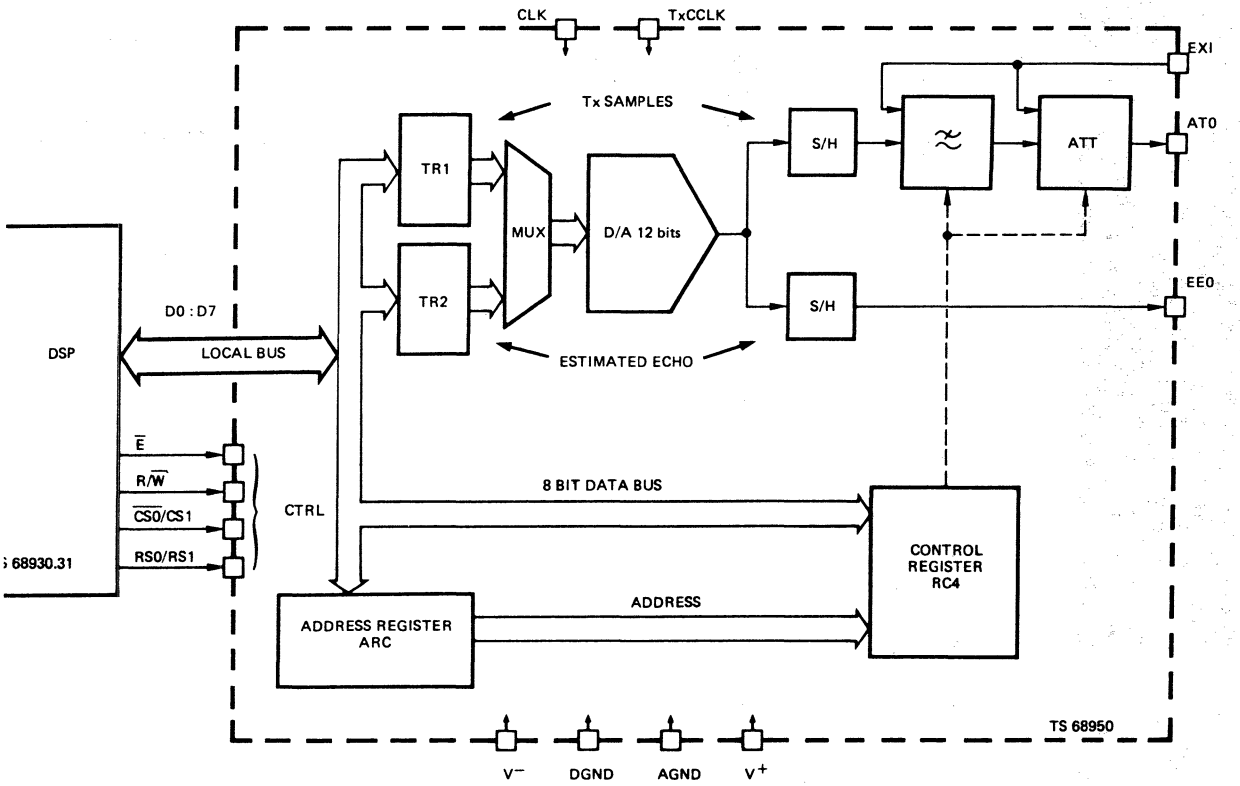
P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE
C SUFFIX
CERAMIC PACKAGE

J SUFFIX
CERDIP PACKAGE

PIN ASSIGNMENT





BLOCK DIAGRAM

PIN DESCRIPTION

Name	No.	Function
D5-D7	1-3	8 bit data bus inputs giving access to Tx, estimated echo, control and address registers. (With pins 20-24).
$\overline{E}$	4	Enable input. Data are strobed on the positive transitions of this input.
R/$\overline{W}$	5	Read/write selection input. Internal registers can be written when R/$\overline{W}$ = 0. Read mode is not used.
$\overline{CS0}$-CS1	6-7	Chip select inputs. The chip set is selected when $\overline{CS0}$ = 0 and CS1 = 1.
RS0-RS1	8-9	Register select inputs. Used to select D/A input registers or control/address registers in the write mode.
DGND	10	Digital ground = 0 V. All digital signals are referenced to that pin.
TEST	11	Test input. Used to reduce testing time. That pin must be connected to DGND in all applications.
V−	12	Negative power supply voltage = −5 V ± 5%
AGND	13	Analog ground = 0 V. Reference point for analog signals.
EXI	14	Programmable analog input tied to filter or attenuator input according to the RC4 register content.
ATO	15	Analog transmit output.
EEO	16	Analog echo cancelling output.
V+	17	Positive power supply voltage: +5 V ± 5%.
CLK	18	1.44 MHz clock input. Used for internal sequencing.
TxCCLK	19	Transmit conversion clock input. Must be derived from CLK.
D0-D4	20-24	See pins No. 1-3.

FUNCTIONAL DESCRIPTION

The TS68950 is a transmit analog interface circuit dedicated to voice-grade MODEMs, telephony and speech applications. The TS68950, the TS68951 (receive analog front-end circuit) and the TS68952 (clock generator) constitute an analog front-end chip set useful for implementation of synchronous MODEMs operating on two or four wires according to the CCITT V.26, V.26 bis, V.27, V.27 bis, V.27 ter and V.29 recommendations or BELL 208 and 209 standards, or in two wires full-duplex according to CCITT V.22, V.22 bis or BELL 212A (split band) and CCITT V.26 ter and V.32 (echo cancelling).
By receiving digital samples from a DSP like the TS68930/31, the TS68950 delivers two analog signals: the transmitted (Tx) signal that will be sent on the line and the estimated echo signal that will be subtracted from the received (Rx) signal on the TS68951 Rx chip.
The digital Tx and estimated echo samples are converted to analog during the low state and the high state of the TxCCLK clock, respectively.

MAIN FUNCTIONS (See block diagram)

- 12 bit digital to analog converter multiplexed on two channels.
- Tx signal sample and hold running with Tx sampling frequency TxCCLK.
- Tx low-pass filter with continuous-time smoothing.
- Programmable attenuator from 0 to −22 dB with 2 dB steps.
- Estimated echo sample and hold running with Tx sampling frequency TxCCLK.

DSP INTERFACE SIGNALS

The TS68950 interfaces to the signal processor via an 8 bit data bus (only used in writing mode), two chip select lines, two register select lines, a read/write line and an enable line.

Data bus (D0-D7) - The write only data lines allow the transfer of data from the DSP to the TS68950. Input buffers are high-impedance devices.

Enable ($\overline{E}$) - The enable pulse ($\overline{E}$) is the basic timing signal that is supplied to the TS68950. All the other signals are referenced to the leading and trailing edges of the $\overline{E}$ pulse.

Read/Write (R/$\overline{W}$) - This signal is generated by the DSP to control the direction of data transfers on the data bus. A low level state on the TS68950 read/write line enables the input buffers and data is transferred from the DSP to the TS68950 on the $\overline{E}$ signal if the circuit has been selected. The device is unselected when a high level signal is applied to the R/$\overline{W}$ pin.

Chip Select ($\overline{CS0}$, CS1) - These two input signals are used to select the chip. $\overline{CS0}$ must be low and CS1 must be high for selection of the device. Data transfers are then performed under the control of the enable and R/$\overline{W}$ signals. The chip select lines must be stable for the duration of the $\overline{E}$ pulse.

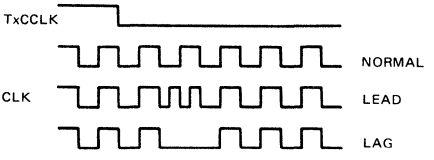
Register Select (RS0, RS1) - The two register select lines are used to access the different registers inside the chip. For instance these two lines are used in conjunction with the internal control register ARC to select a particular register RC4. The register select lines must be stable when the $\overline{E}$ signal is low.

CLOCK INTERFACE BETWEEN TS68950 AND TS68952

The TS68950 receives two clock lines from the Clock Generator TS68952.

Master clock sequencing (CLK)

The typical frequency is 1.44 MHz but the recurrence frequency must be an exact multiple of the terminal clock frequency. The Tx DPLL included in the clock generator circuit (TS68952) operates by adding or subtracting pulses to a 2.88 MHz internal clock. This corresponds to phase leads or phase lags of about 350 ns duration. To ensure correct device operation, clock synchronization must be done immediately after the negative-going transition of TxCCLK clock.



Transmit Conversion Clock (TxCCLK)

The conversion clock TxCCLK must be derived from the master clock CLK. Three nominal values are possible: 9.6 kHz, 8 kHz and 7.2 kHz. 9.6 kHz is the highest allowable frequency. To run properly the TxCCLK clock must be a submultiple of CLK/5:

TxCCLK x 5 x N = CLK (with N integer)

This is ensured when using the TS68952 clock generator.
The sampling clock of the switched capacitor filter section is obtained by dividing the CLK frequency by five and performing internal synchronization on the leading edges of TxCCLK.
The Tx samples are converted from digital to analog during the low state of TxCCLK. The estimated echo samples are converted during the high state of TxCCLK.

INTERNAL CONTROLS

Power-on

The chip contains internal power-on reset logic to initialize the RC4 control register in order to avoid undesirable signal transmission on the telephone line.

Internal addressing

RS0	RS1	Access	320 ns cycle number
0	0	TR1 transmitted sample register	2
0	1	TR2 estimated echo sample register	2
1	0	ARC address register	1
1	1	RC4 control register (if addressed by ARC)	1

Sample registers (TR1 and TR2)

TR1 is the transmitted sample register and TR2 the estimated echo sample register. TR1 and TR2 store two's complement 12 bit data (DAC0 to DAC11). As indicated below, writing each sample requires two cycles.

	D7	D6	D5	D4	D3	D2	D1	D0
First cycle	DAC 3	DAC 2	DAC 1	DAC 0	X	X	X	X
Second cycle	DAC 11	DAC 10	DAC 9	DAC 8	DAC 7	DAC 6	DAC 5	DAC 4

An internal flip-flop is used to select the first or the second byte. It advances one count on the positive-going edge of the $\overline{E}$ pulse when the sample registers are selected ($\overline{CS0}$ = 0, CS1 = 1 and RS0 = 0). When the sample registers are disabled, the latch is reset on any $\overline{E}$ positive-going edge. Both TR1 and TR2 registers are sampled by the DAC on the falling edge of TxCCLK. Therefore their contents must remain stable during this edge.

Control register (RC4)

The RC4 control register has two different functions. Its four most significant bits give the transmit attenuator gain following the table below.

D7	D6	D5	D4	D3	D2	D1	D0	RC4 REGISTER
ATT 4	ATT 3	ATT 2	ATT 1	–	EM2	EM1	–	Attenuation (dB)
0	0	0	0					0
0	0	0	1					2
0	0	1	0					4
0	0	1	1					6
0	1	0	0					8
0	1	0	1					10
0	1	1	0					12
0	1	1	1					14
1	0	0	0					16
1	0	0	1					18
1	0	1	0					20
1	0	1	1					22
1	1	0	0					Infinite
1	1	0	1					Infinite
1	1	1	0					Infinite
1	1	1	1					Infinite

Depending on the EM1 and EM2 states in the RC4 register, the programmable analog input (EXI) can be connected to the filter input or to the transmit attenuator input.

D7	D6	D5	D4	D3	D2	D1	D0	RC4 REGISTER
ATT 4	ATT 3	ATT 2	ATT 1	–	EM2	EM1	–	EXI INPUT
					0	0		disabled
					0	1		transmit filter input
					1	0		transmit attenuator input
					1	1		disabled

Following power-up, all RC4 bits are preset at one, EXI input is disabled and the transmit signal is cancelled.

D0 and D3 bits are not used in the RC4 register.

Address register (ARC)

The address register stores 3 bits (D5, D6 and D7). Among the 8 possible addresses, only one is used inside the TS68950 (RC4 address).

	D7	D6	D5	D4	D3	D2	D1	D0
RC 4	0	1	1	X	X	X	X	X

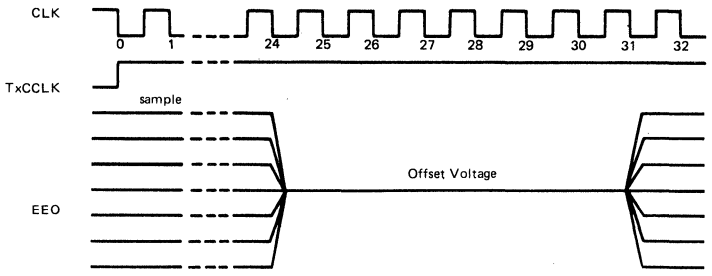
X: don't care

The address of the ARC register is automatically increased by one each time the control register is accessed. This allows indirect or cyclical addressing to RC4.

EEO OUTPUT WAVEFORM

The EEO output is not valid during S/H sampling. The output presents at this time the S/H offset voltage.
This offset voltage appears at the 24th CLK period after rise transition of TxCCLK and disappears at the 31th.

Waveform



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
DGND digital ground to AGND analog ground		− 0.3 to + 0.3	V
V^+ supply voltage to DGND or AGND ground		− 0.3 to + 7	V
V^- supply voltage to DGND or AGND ground		− 7 to + 0.3	V
Voltage at any digital input or output	V_I	DGND − 0.3 to V^+ + 0.3	V
Voltage at any analog input or output	V_{in}	V^- − 0.3 to V^+ + 0.3	V
Analog output current	I_{out}	− 10 to + 10	mA
Power dissipation	P_{tot}	500	mW
Operating temperature range	t_{amb}	0 to + 70	°C
Storage temperature range	t_{stot}	− 65 to + 150	°C
Pin temperature (soldering 10 s.)	t_{sold}	+ 260	°C

Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. Standard CMOS handling procedures should be employed to avoid possible damage to device.

ELECTRICAL OPERATING CHARACTERISTICS

Characteristic	Symbol	Min	Typ	Max	Unit
Positive Supply Voltage	V^+	4.75	5	5.25	V
Negative Supply Voltage	V^-	− 5.25	− 5.0	− 4.75	V
V^+ Operating current	I^+	–		15	mA
V^- Operating current	I^-	− 15		–	mA

D.C. AND OPERATING CHARACTERISTICS

Unless otherwise noted, electrical characteristics are specified over the operating range. Typical values are given for $V^+ = +5$ V, $V^- = -5$ V and $t_{amb} = 25°C$.

DIGITAL INTERFACE

Parameter	Symbol	Min	Typ	Max	Unit
Input low level voltage	V_{IL}			0.8	V
Input high level voltage	V_{IH}	2.2			V
Input low level current $DGND < V_I < V_{ILmax}$	I_{IL}	− 10		10	μA
Input high level current $V_{IHmin} < V_I < V^+$	I_{IH}	− 10		10	μA

ANALOG INTERFACE, EXI PROGRAMMABLE INPUT

Parameter	Symbol	Min	Typ	Max	Unit
Input voltage swing	V_{in}	− 2.5		+ 2.5	V
Input current (Input Tx filter selected)	I_{in}	− 10		+ 10	μA
Input capacitance (Input ATT selected) f < 50 kHz f > 50 kHz	C_{in}			 50 20	 pF pF
Input resistance (Input ATT selected)	R_{in}	20			kΩ

ANALOG INTERFACE, ATO TRANSMIT OUTPUT

Parameter	Symbol	Min	Typ	Max	Unit
Output DC offset	V_{os}	− 250		+ 250	mV
Load capacitance	C_L			50	pF
Load resistance	R_L	1200			Ω
Output voltage swing $R_L > 1200\ \Omega$ and $C_L < 50$ pF	V_{out}	− 2.5		+ 2.5	V
Output resistance	R_{out}			5	Ω

ANALOG INTERFACE, EEO ESTIMATED ECHO OUTPUT

Parameter	Symbol	Min	Typ	Max	Unit
Output DC offset	V_{os}	− 100		+ 100	mV
Load capacitance	C_L			50	pF
Load resistance	R_L	10			kΩ
Output voltage swing $R_L > 10$ kΩ and $C_L < 50$ pF	V_{out}	− 2.5		+ 2.5	V
Output resistance	R_{out}	350	500	650	Ω

DAC TRANSFER CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Unit
Converter resolution			12		Bit
Nominal output peak to peak amplitude	V_{out} (max)		5.0		V
Least significant bit amplitude	LSB		1.2		mV
Integral linearity error		− 1		+ 1	LSB
Differential linearity error		− 0.7		+ 0.7	LSB

TRANSMIT FILTER TRANSFER CHARACTERISTICS (see annexe 1)

Parameter	Symbol	Min	Typ	Max	Unit
Absolute passband gain at 1 kHz	G_{AR}		0		dB
Gain relative to gain at 1 kHz without sin x/x correction of DAC sampling	G_{RR}				
Below 3100 Hz		− 0.5		0.2	dB
3200 Hz		− 3			dB
4000 Hz				− 36	dB
5000 Hz to 12000 Hz				− 46	dB
12000 Hz and above				− 50	dB
Absolute delay 600 Hz to 3000 Hz	D_{AR}	160		680	μs

ATTENUATOR TRANSFER CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Unit
Absolute gain at 0 dB nominal value	A_{TT}		0		dB
Attenuation relative to nominal value	R_{AT}	− 5.0		+ 0.5	dB
Maximum attenuation	B_{AT}	40			dB

GENERAL TRANSFER CHARACTERISTICS (from DATA BUS to ATO)

Parameter	Symbol	Min	Typ	Max	Unit
ATO absolute gain at 1 kHz	G_{AX}	− 0.5	0	+ 0.5	dB
ATO psophometric noise				100	μV
ATO positive power supply rejection ratio. V_{ac} = 200 mVpp f = 1 kHz			40		dB
ATO negative power supply rejection ratio. V_{ac} = 200 mVpp f = 1 kHz			40		dB
Signal to harmonic distorsion ratio (psophometric band)		60			dB

GENERAL TRANSFER CHARACTERISTICS (from DATA BUS to EEO)

Parameter	Symbol	Min	Typ	Max	Unit
EEO absolute gain at 1 kHz	G_{AX}	− 0.5	0	+ 0.5	dB
EEO psophometric noise				100	μV
EEO positive power supply rejection ratio. V_{ac} = 200 mVpp f = 1 kHz			40		dB
EEO negative power supply rejection ratio. V_{ac} = 200 mVpp f = 1 kHz			40		dB

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident number	Characteristic	Symbol	Min	Max	Unit
1	Cycle time	t_{cyc}	320		ns
2	Pulse width, $\overline{E}$ low level	t_{WEL}	180		ns
3	Pulse width, $\overline{E}$ high level	t_{WEH}	100		ns
4	Clock rise and fall time	t_r, t_f		20	ns
5	Control signal hold time	t_{HCE}	10		ns
6	Control signal set-up time	t_{SCE}	40		ns
7	Input data set-up time	t_{SDI}	120		ns
8	Input data hold time	t_{HDI}	10		ns

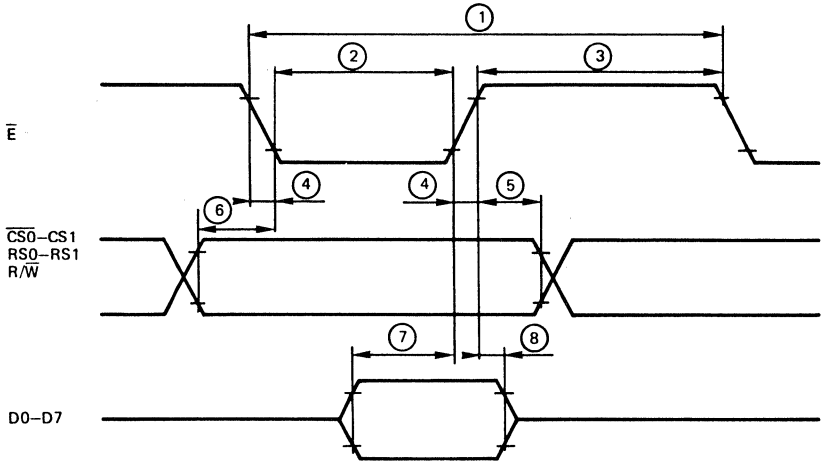


FIGURE 1 – BUS TIMING

Notes :

1. Voltage levels shown are VL < 0.4 V, VH > 2.4 V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.2 V, unless otherwise specified.

CLOCK TIMING CHARACTERISTICS

Ident number	Characteristic	Symbol	Min	Typ	Max	Unit
1	CLK clock period	P_C		695		ns
2	CLK phase leading clock period	P_{CL}		348		ns
3	CLK low level width	t_{WCL}	150			ns
4	CLK high level width	t_{WCH}	150			ns
5	CLK rise and fall time	t_{RC}, t_{FC}			100	ns
6	TxCCLK rise and fall time	t_{RT}, t_{FT}			100	ns
7	TxCCLK delay time	t_{DC}	20		130	ns

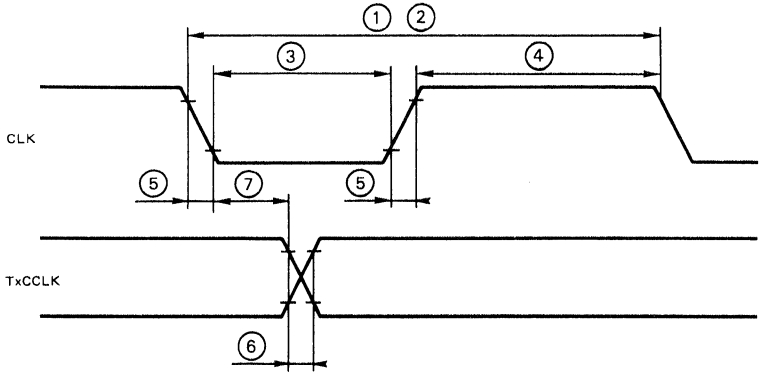
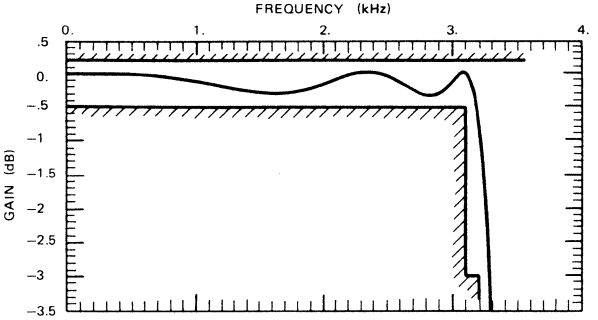
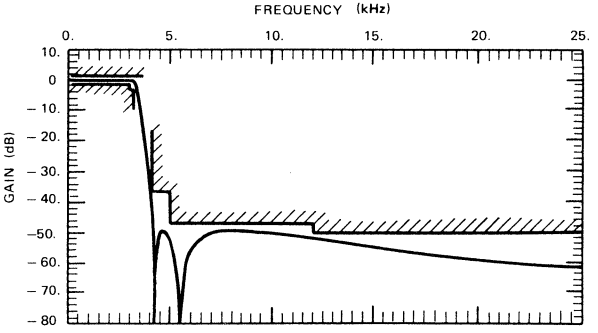


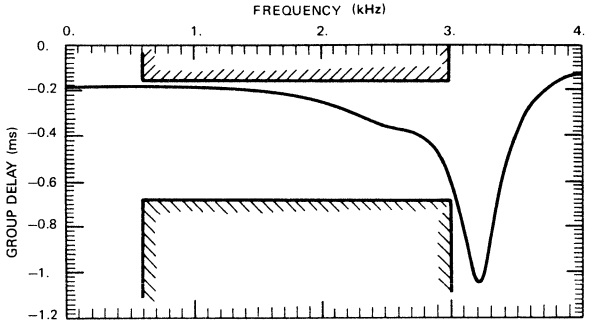
FIGURE 2 – CLOCK TIMING



TRANSMIT LOW-PASS FILTER TYPICAL RESPONSE AND LIMITS CHART



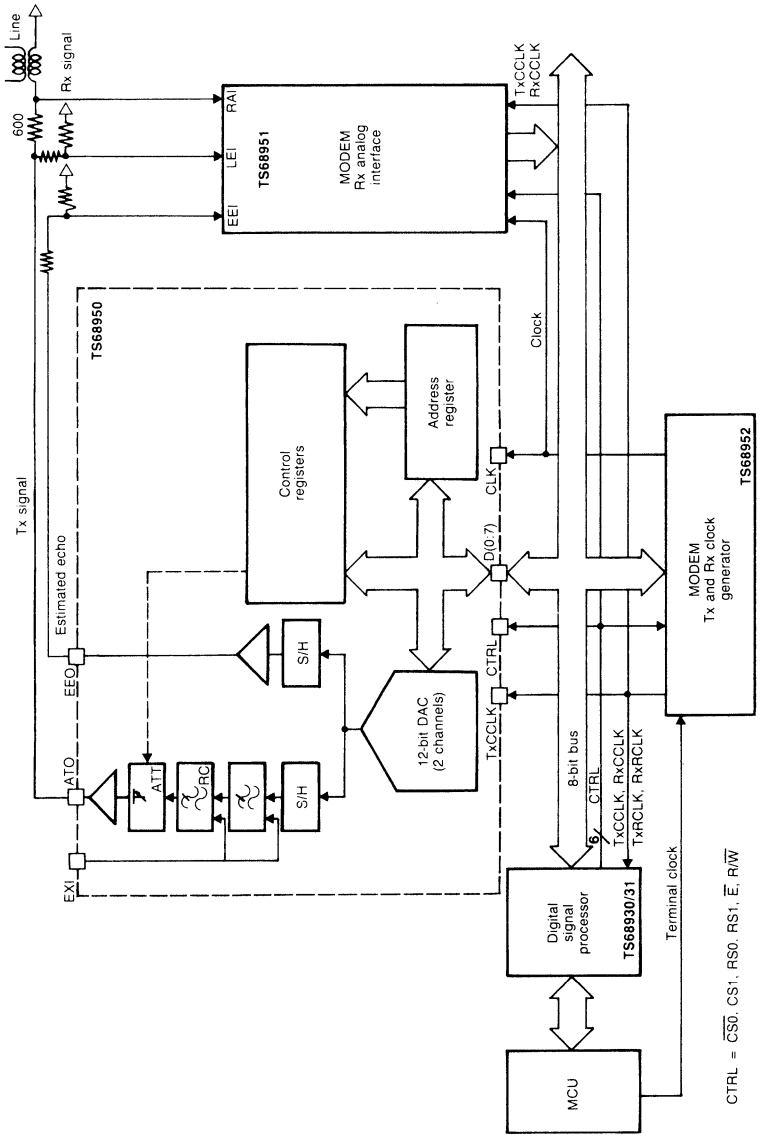
TRANSMIT LOW-PASS FILTER TYPICAL RESPONSE AND LIMITS CHART



TRANSMIT LOW-PASS FILTER TYPICAL GROUP DELAY AND LIMITS CHART

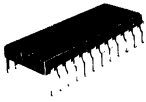
APPENDIX 1

TYPICAL APPLICATION

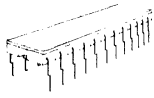


PHYSICAL DIMENSIONS

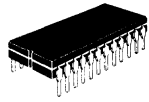
CB-68



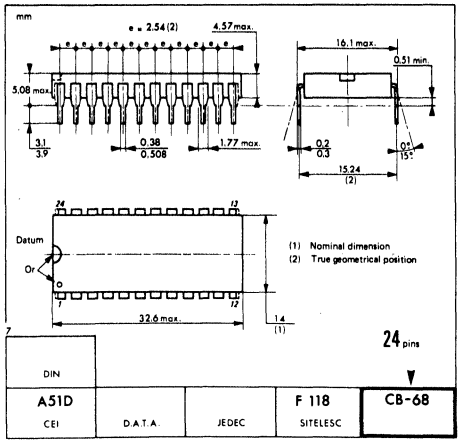
P SUFFIX
PLASTIC PACKAGE



ALSO AVAILABLE
C SUFFIX
CERAMIC PACKAGE



J SUFFIX
CERDIP PACKAGE



THOMSON COMPONENTS
MOSTEK

TS68951

MODEM RECEIVE ANALOG INTERFACE

COMMUNICATIONS PRODUCTS

DATA SHEET

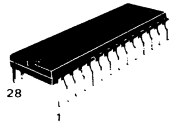
CMOS
MODEM RECEIVE ANALOG INTERFACE

The TS68951 is the receive section of a MODEM analog front-end. The MODEM consists of TS68950/51/52 analog front-end chip sets and TS68930/31 digital signal processor; it is able to run voice-grade applications, which conforms to CCITT V.22/BIS, V.26/TER, V.27, V.29, V.32 and V.33 recommendations as well as BELL 212A, 208 and 209 standards.

Main features

- Programmable band-pass filter.
- Back channel rejection filter (selected by programming)
- Reconstruction filter (selected by programming)
- Continuous-time anti-aliasing and smoothing filters
- Programmable gain amplifier (from 0 dB to 46.5 dB with 1.5 dB steps)
- 12 bit A/D converter with asynchronous multiplexing of two plesiochronous channels (one channel for echo cancellation)
- Carrier level detector with programmable threshold.
- Digital interface: 8 bit bi-directional data bus, 6 bit control bus.
- Dual power supplies +5 V and −5 V
- Designed to operate with TS68950 transmit unit and TS68952 clock generator.

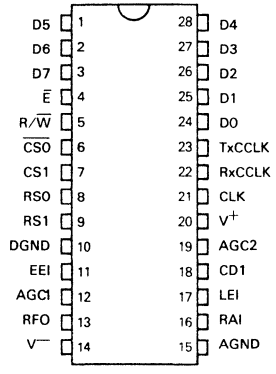
CASE CB-132



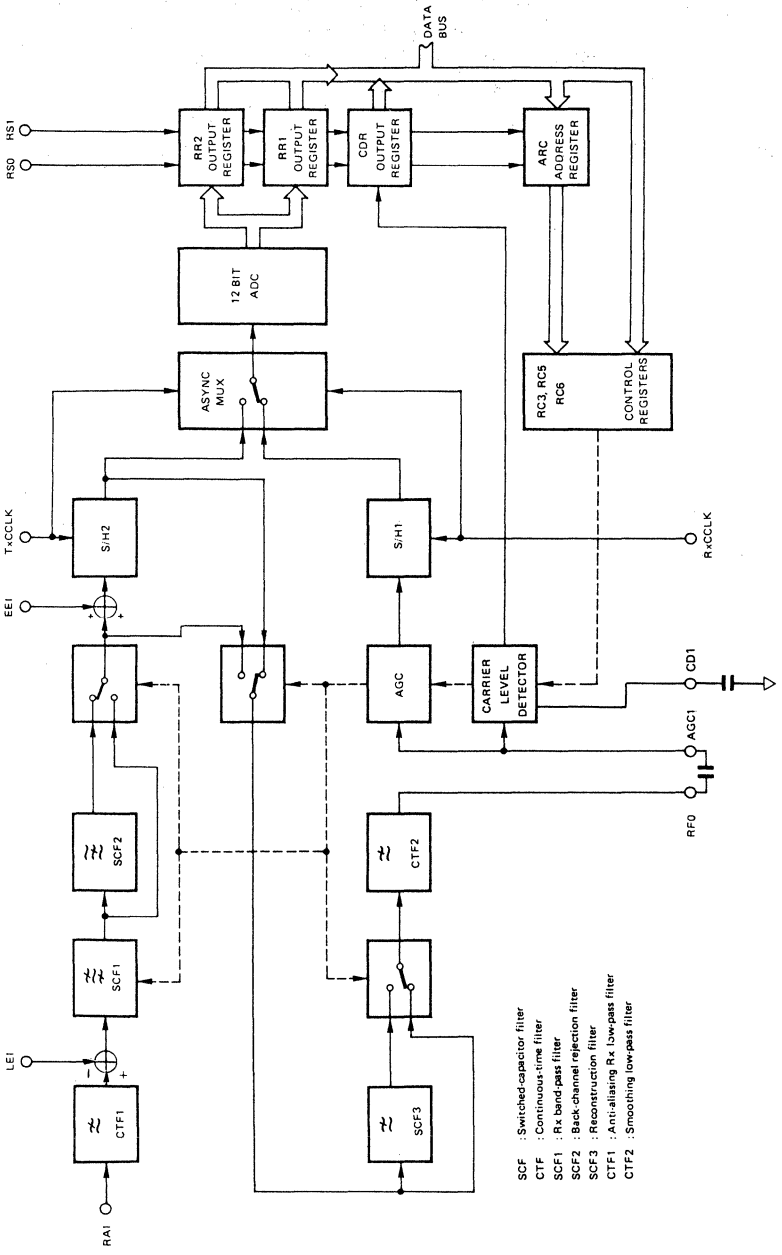
P SUFFIX
PLASTIC PACKAGE

ALSO AVAILABLE
C SUFFIX
CERAMIC PACKAGE
J SUFFIX
CERDIP PACKAGE

PIN ASSIGNMENT



BLOCK DIAGRAM



PIN DESCRIPTION

Name	No.	Description
D5-D7	1-3	Data bus
$\overline{E}$	4	Enable input. Enables selection inputs. Active on a low level for read operation. Active on a positive edge for write operation.
R/$\overline{W}$	5	Read/write Selection input. Read operation is selected on a high level. Write operation is selected on a low level.
$\overline{CS0}$-CS1	6-7	Chip select inputs. The chip set is selected when $\overline{CS0} = 0$ and CS1 = 1.
RS0-RS1	8-9	Register select inputs. Select the register involved in a read or write operation.
DGND	10	Digital ground. All digital signals are referenced to this pin.
EEI	11	Estimated echo input. When operating in echo cancelling mode, this signal is added to the reception band-pass filter output.
AGC1	12	Analog input of the automatic gain control amplifier and of the carrier level detector.
RFO	13	Reception filter analog output. Designed to be connected to AGC1 input through a 1 μF non polarised capacitor.
V $-$	14	Negative power supply. V $-$ = $-$ 5 V $\pm$ 5%.
AGND	15	Analog ground. All analog signals are referenced to this pin.
RAI	16	Receive analog input. Analog input tied to the transmission line.
LEI	17	Local echo input. Analog input subtracted from the receive anti-aliasing filter output.
CD1	18	This pin must be connected to the analog ground through a 1 μF non polarised capacitor, in order to cancel the offset voltage of the carrier level detector amplifier.
AGC2	19	This pin must be connected to the analog ground through a 1 μF non polarised capacitor, in order to cancel the offset voltage of the AGC amplifier.
V+	20	Positive power supply V$^+$ = + 5 V $\pm$ 5%.
CLK	21	Master clock input. Nominal frequency 1.44 MHz.
RxCCLK	22	Receive conversion clock.
TxCCLK	23	Transmit conversion clock.
D0-D4	24-28	Data bus.

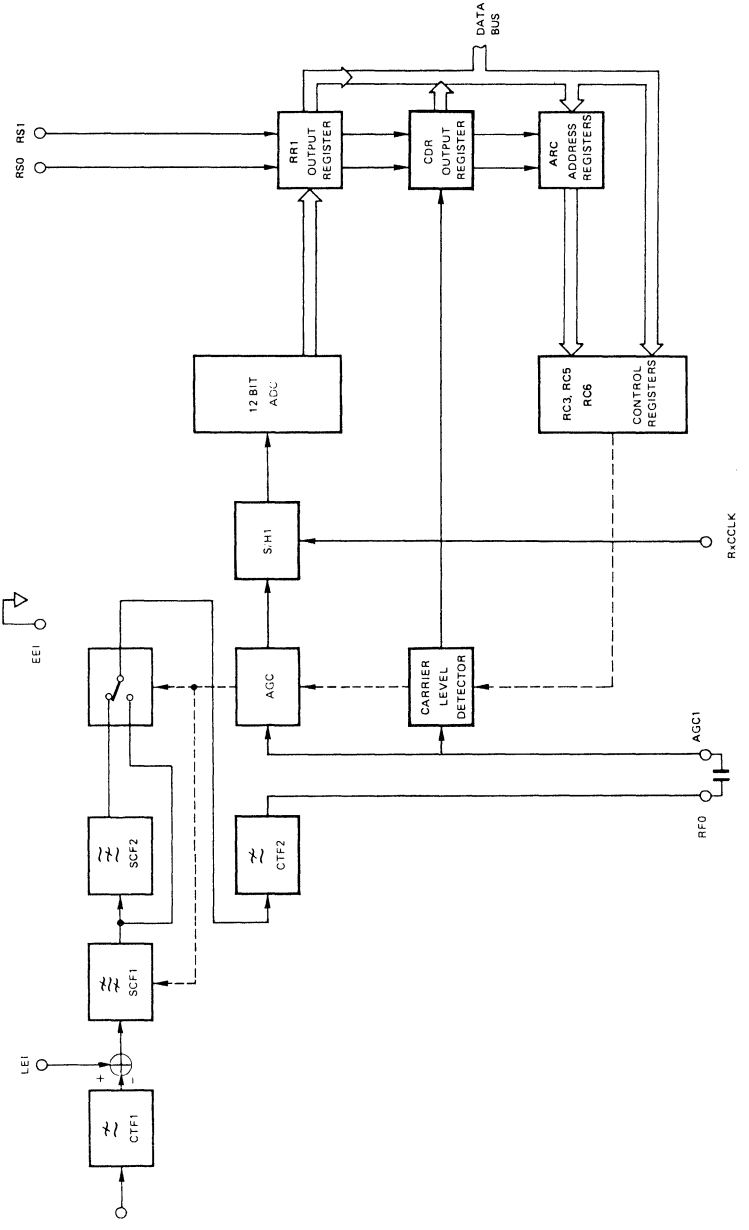


FIGURE 1 – FOUR-WIRE OR TWO-WIRE HALF DUPLEX AND TWO-WIRE BAND-SPLIT ANALOG SIGNAL TREATMENT

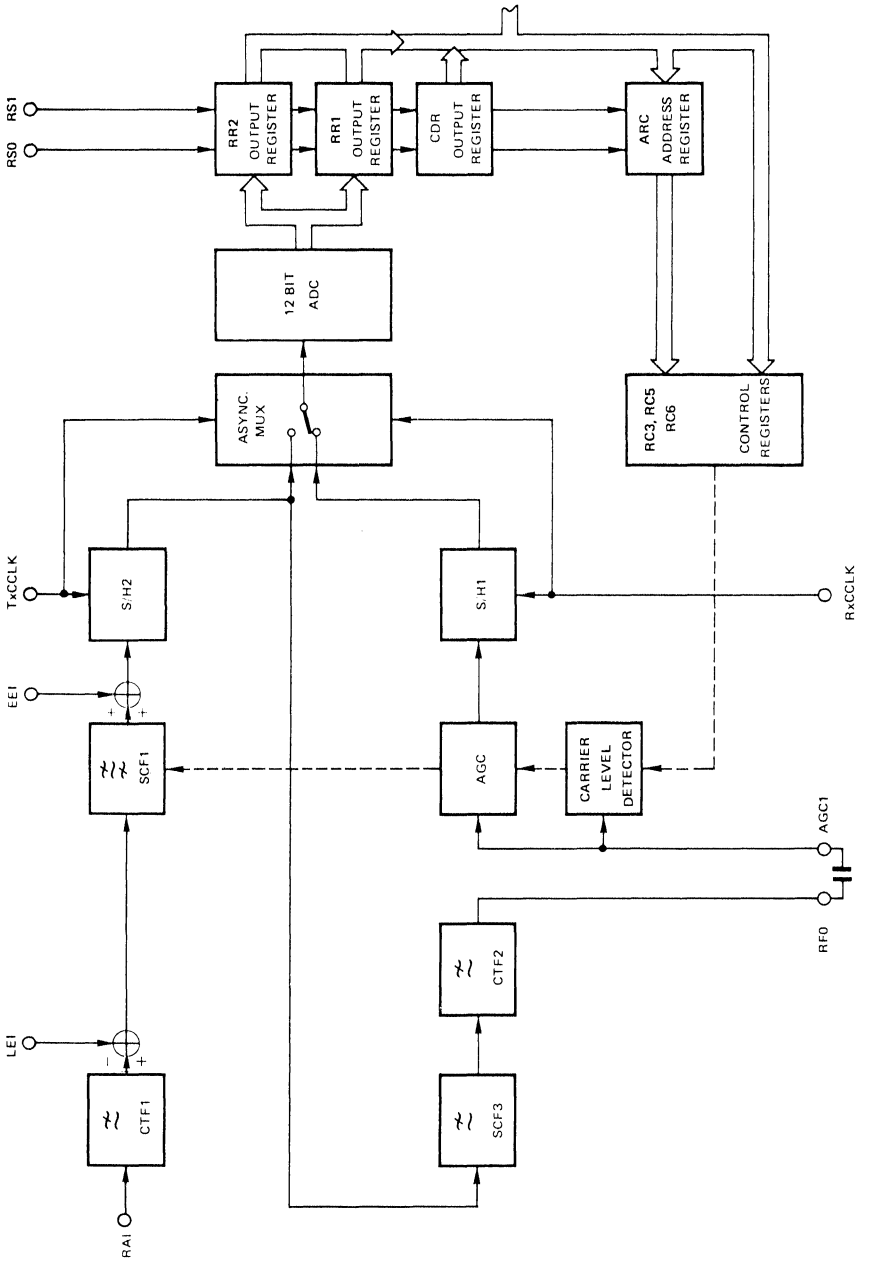


FIGURE 2 – TWO-WIRE ECHO CANCELLING ANALOG SIGNAL TREATMENT

FUNCTIONAL DESCRIPTION

The TS68951 is a receive analog interface for voice-grade MODEM. It is able to perform the receive interface function for three types of synchronous MODEM:

- Four-wire or two-wire half duplex MODEM.
- Two-wire full duplex band-split MODEM.
- Two-wire full duplex echo cancelling MODEM.

Four-wire or two-wire half duplex MODEM and two-wire band-split MODEM

In these modes of operation, EEI input must be tied to the analog ground. The analog signal treatment of receive input is shown in figure 1.
Programming requirements:

- Band-pass filter cut-off frequencies.
- Back channel rejection filter (presence or absence according to the application).
- SCF1 or SCF2 output as input of CTF2.
- AGC gain.
- Carrier level detector threshold.

The receive samples are coded at RxCCLK rate and can be read from receive register (RR1).

Two-wire echo cancelling MODEM

This mode of operation uses the full capabilities of the TS68951. The analog treatment of receive input is shown in figure 2. The echo cancelling operation is achieved by means of subtraction of the LEI signal from the output of CTF1 duplexer and addition of the EEI signal to the output of SC1.
After the local echo reduction by the duplexer the resultant signal consists of the receive signal plus the echo signal generated by the transmission line mismatch: this undesirable signal is then cancelled at the output of the Rx band-pass filter.
Programming requirements:

- Band-pass filter cut-off frequencies.
- SCF1 output as input of S/H2.
- Output of S/H2 as input of SCF3 and output of SCF3 as input of CTF2.
- AGC gain.
- Carrier level detector threshold.

Residual signal samples from S/H2 output are coded at TxCCLK rate and can be read from receive register 2 (RR2), hence the signal processor may correlate them with the transmit samples to update the coefficients of the filter that generates the estimated echo.
The receive signal samples are coded at RxCCLK rate and can be read from receive register 1 (RR1).

FUNCTIONAL SPECIFICATIONS

Bus and Registers Control

For any operation involving bus and registers, the chip select bits $\overline{CS0}$ and CS1 must be valid ($\overline{CS0}$ =0 and CS1 = 1).
The seven internal registers are divided in four write only registers and three read-only registers.

Write operation

There are three control registers (RC3, RC5, RC6) and one address register (ARC) which can be written; but only ARC can be directly addressed.
The control registers are indirectly addressed by the word contained in ARC according to table 1.

Addressed control register	Word contained in ARC							
	D7	D6	D5	D4	D3	D2	D1	D0
RC3	0	1	0	X	X	X	X	X
RC5	1	0	0	X	X	X	X	X
RC6	1	0	1	X	X	X	X	X

X: don't care

TABLE 1

When a write operation is selected (refer to table 3) the data present on the bus are strobed on a positive edge of $\overline{E}$ and the content of ARC is incremented.

Note: Addresses of RC3 and RC5 are separated by two increments.

Read operation

There are two 12 bit receive registers (RR1, RR2) and a 1 bit carrier detector register (CDR).

RR2 contains the coded samples of the residual signal and RR1 the coded samples of the receive signal.

The active bit of CDR is D7: D0 to D6 are forced to 0.

When the RMS value of CTF2 output is greater than the programmed threshold, bit 7 of CDR is set. The nominal response time of the carrier detector to a signal settlement or removal is 1.78 ms.

When a read operation is selected (refer to table 3) the data are sent to the bus on a low level of $\overline{E}$; a high level on $\overline{E}$ sets the output bus drivers in a high impedance state.

As the bus has only 8 bits, the content of RR1 or RR2 must be read in two cycles. The four less significant bits are transferred in the first cycle and the eight most significant bits are transferred in the second cycle according to the format, table 2.

	D7	D6	D5	D4	D3	D2	D1	D0
First cycle	RRx3	RRx2	RRx1	RRx0	0	0	0	0
Second cycle	RRx11	RRx10	RRx9	RRx8	RRx7	RRx6	RRx5	RRx4

TABLE 2

An internal latch selects the first or the second byte and is automatically incremented on a positive edge of $\overline{E}$ when one of the receive registers is addressed. This latch is not reset at power-on, so it needs to be reset before the first read operation: reset occurs on any positive edge of $\overline{E}$ for any operation, provided none of the receive registers is addressed; the first byte is selected when reset.

R/$\overline{W}$	RS0	RS1	Operation
0	1	1	Write control register addressed by ARC
0	1	0	Write address register (ARC)
1	0	1	Read receive register 2 (RR2) (Residual signal sample)
1	0	0	Read receive register 1 (RR1) (Receive signal sample)
1	1	0	Read carrier detector register (CDR)

TABLE 3

RR1 and RR2 output code:
The output code is a 2's complement delivering values from - 2048 up to + 2047. Since the converter codes voltage between $-V_{ref}$ and $+V_{ref}$, the theoretical decision voltage corresponding to code C can be computed as follows:

$$V_c = \frac{2C+1}{4095} V_{ref}$$

where V_{ref} is the reference voltage of the A/D converter, V_{ref} nominal value is 2.5 V and C is the algebraic value of code C.

Example:
Assume the output code is the hexadecimal value $8B1; the algebraic value of this code C = — 1871 therefore V_c = — 2.283 V.

CONTROL REGISTERS DESCRIPTION

Power-on
The control registers are not initialized at power on; they must be initialized from program before reading any word from the output registers.

Register RC3
The content of RC3 sets the — 3 dB cut-off frequencies of SCF1 receive band-pass filter, determines the presence or the absence of SCF2 back channel rejection filter and of SCF3 reconstruction filter, and selects receive signal path to the second filtering section ; without echo-cancelling the output of SCF1 or SCF2 is selected ; with echo-cancelling the output of S/H2 is selected.

The band pass filter consists of a 5th order elliptic low-pass filter and of a 2nd order high-pass filter whose cut off frequencies can be programmed by (LP1, LP2) and (HP1, HP2) respectively, (refer table 4).

The rejection filter is present when REJ bit is high.

The reconstruction filter is present when REC bit is high.

S/H2 output is selected when S/A bit is high.

D7 HP2	D6 HP1	D5 LP2	D4 LP1	D3 REJ	D2 S/A	D1 REC	D0	RC3 REGISTER		
								LOW-PASS FILTER		
								Sampling frequency (kHz)	— 3 dB Cut-off freq. (Hz)	
		0 0 1 1	0 1 0 1				X X X X	72 144 288 288	800 1600 3200 3200	
								HIGH-PASS FILTER		
								Sampling frequency (kHz)	— 3 dB Cut-off freq. (Hz)	
0 1 1	X 0 1			0 0 0			X X X	36 72 144	250 500 1600	
								HIGH-PASS AND REJECTION FILTER		
								Sampling freq. (kHz)	— 3 dB Cut-off freq. (Hz)	Rejected band (Hz)
1 1	0 1			1 1			X X	72 144	800 2200	370-470 800-1600
								S/H2 SELECTION		
					0 1		X X	Deselected Selected		
								RECONSTRUCTION FILTER SELECTION		
						0 1	X X	Deselected Selected	(Sampling frequency = 288kHz)	

TABLE 4

X: don't care

Register RC5

The content of RC5 sets the gain of the AGC amplifier between 0 dB and 46.5 dB with 1.5 dB steps.

Note: The AGC loop control is performed by the signal processor.

D7	D6	D5	D4	D3	D2	D1	D0	RC5
								AGC gain (dB)
0	0	0	0	0	X	X	X	0
0	0	0	0	1	X	X	X	1.5
0	0	0	1	0	X	X	X	3
0	0	0	1	1	X	X	X	4.5
0	0	1	0	0	X	X	X	6
0	0	1	0	1	X	X	X	7.5
0	0	1	1	0	X	X	X	9
0	0	1	1	1	X	X	X	10.5
0	1	0	0	0	X	X	X	12
0	1	0	0	1	X	X	X	13.5
0	1	0	1	0	X	X	X	15
0	1	0	1	1	X	X	X	16.5
0	1	1	0	0	X	X	X	18
0	1	1	0	1	X	X	X	19.5
0	1	1	1	0	X	X	X	21
0	1	1	1	1	X	X	X	22.5
1	0	0	0	0	X	X	X	24
1	0	0	0	1	X	X	X	25.5
1	0	0	1	0	X	X	X	27
1	0	0	1	1	X	X	X	28.5
1	0	1	0	0	X	X	X	30
1	0	1	0	1	X	X	X	31.5
1	0	1	1	0	X	X	X	33
1	0	1	1	1	X	X	X	34.5
1	1	0	0	0	X	X	X	36
1	1	0	0	1	X	X	X	37.5
1	1	0	1	0	X	X	X	39
1	1	0	1	1	X	X	X	40.5
1	1	1	0	0	X	X	X	42
1	1	1	0	1	X	X	X	43.5
1	1	1	1	0	X	X	X	45
1	1	1	1	1	X	X	X	46.5

TABLE 5

X: don't care

Register RC6

The content of RC6 sets the carrier level detector threshold. (Refer to table 6).

The threshold values are grouped by pair; values belonging to each pair have 2.5 dB separation which allows the signal processor to perform software hysteresis.

D7	D6	D5	D4	D3	D2	D1	D0	RC6
								Threshold (dBm)
0	0	0	X	X	X	X	X	— 29.85
0	0	1	X	X	X	X	X	— 27.35
0	1	0	X	X	X	X	X	— 36.65
0	1	1	X	X	X	X	X	— 34.15
1	0	0	X	X	X	X	X	— 46.75
1	0	1	X	X	X	X	X	— 44.25
1	1	0	X	X	X	X	X	— 46.75
1	1	1	X	X	X	X	X	— 44.25

TABLE 6

X don't care

CLOCK

The master clock CLK, the receive conversion clock (RxCCLK) and the transmit conversion clock (TxCCLK) are generated in the TS68952 clock generator. There are three possible frequencies for the conversion clocks: 7.2 kHz, 8 kHz and 9.6 kHz.

The frequency of RxCCLK and TxCCLK is controlled by two independant Digital Phase Locked Loops (DPLL). TxCCLK can be synchronised on an external Terminal Clock (TxSCLK) or on the Rx bit rate clock ; in these cases 350 ns discrete phase shifts occurs on CLK and TxCCLK synchronously with TxCCLK negative edge with a repetition rate of 600 Hz, 800 Hz or 1000 Hz according to the programmation of RC1 control register in the TS68952.

AGC and CLD AMPLIFIERS

The AGC consists of two cascaded amplifiers A1 and A2, fig. 3. AC coupling is obtained from C1 and C2 external capacitors. C2 can be used as an auxiliary input for performing an analog loop located after echo cancellation. The carrier level detector (CLD) amplifier A3 also needs an external capacitor C3.

A/D CONVERSION

The A/D converter is a 12 bit resolution, 8 bit minimum integral linearity, monotonic converter. The input voltage ranges from — 2.5 V to + 2.5 V; and the conversion time is better than 50 μs.

ASYNCHRONOUS MULTIPLEXING

Samples on the output of S/H1 and S/H2 are converted respectively at RxCCLK frequency and TxCCLK frequency. Since RxCCLK and TxCCLK are plesiochronous, the order of conversion is determined by an asynchronous logic. The output register RR1 and RR2 are respectively loaded on the negative edge of RxCCLK and TxCCLK.

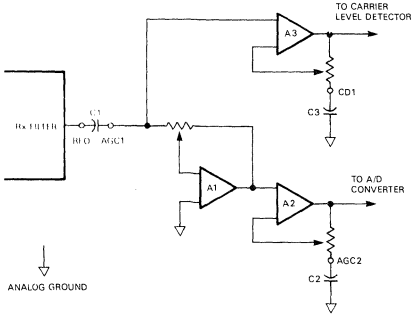


FIGURE 3. Rx AMPLIFIERS SCHEMATIC

ELECTRICAL SPECIFICATIONS

The electrical specifications are given for operating temperature range (0°C, 70°C).

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply voltage between V^+ and AGNC or DGND		− 0.3 to + 7	V
Supply voltage between V^- and AGND or DGND		− 7 to + 0.3	V
Voltage between AGND and DGND		− 0.3 to + 0.3	V
Digital input voltage		DGND−0.3 to V^++0.3	V
Digital output voltage		DGND−0.3 to V^++0.3	V
Digital output current		− 20 to + 20	mA
Analog input voltage		V_{CC}−0.3 to V^++0.3	V
Analog output voltage		V_{CC}−0.3 to V^++0.3	V
Analog output current		− 10 to + 10	mA
Power dissipation		500	mW
Operating temperature	T_{oper}	0 to + 70	°C
Storage temperature	T_{stg}	−65 to + 150	°C

POWER SUPPLIES

DGND, AGND = 0 V

Characteristic	Symbol	Min	Typ	Max	Unit
Positive power supply	V^+	4.75	—	5.25	V
Negative power supply	V^-	− 5.25	—	− 4.75	V
Positive supply current (receive signal level 0 dBm)	I^+	—	—	20	mA
Negative supply current (receive signal level 0 dBm)	I^-	− 20	—	—	mA

DIGITAL INTERFACE

Control inputs

Voltages referenced to DGND = 0 V

Characteristic	Symbol	Min	Typ	Max	Unit
Low level input voltage	V_{IL}	—	—	0.8	V
High level input voltage	V_{IH}	2.2	—	—	V
Low level input current DGND < V_I < 0.8 V	V_{IL}	− 10	—	10	μA
High level input current 2.2 V < V_I < V^+	V_{IH}	− 10	—	10	μA

DATA BUS

Voltages referenced to DGND = 0 V

Characteristic	Symbol	Min	Typ	Max	Unit
Low level input voltage	V_{IL}	—	—	0.8	V
High level input voltage	V_{IH}	2.2	—	—	V
Low level output voltage (I_{OL} = 2.5 mA)	V_{OL}	—	—	0.4	V
High level output voltage (I_{OL} = 2.5 mA)	V_{OH}	2.4	—	—	V
High impedance output current (when $\overline{E}$ is high and DGND < V_I < V^+)	I_{OZ}	− 50	—	50	μA

ANALOG INTERFACE

All voltages referenced to AGND = 0 V

Characteristic	Symbol	Min	Typ	Max	Unit
Input voltage EEI,LEI,RAI	V_{in}	— 2.5	—	2.5	V
Input current EEI,LEI,RAI (— 2.5 V < V_{in} < 2.5 V)	I_{in}	— 1	—	1	μA
Input resistance AGC1, AGC2	R_{in}	1.5	—	—	kΩ
Input resistance CD1	R_{in}	0.7	—	—	kΩ
Output voltage RFO CL = 50 pF, RL = 1 kΩ	V_{out}	— 2.5	—	2.5	V
Output resistance RFO	R_{out}	—	—	2	Ω
Load resistance RFO	R_L	1	—	—	kΩ
Load capacitance RFO	C_L	—	—	50	pF

BUS TIMING CHARACTERISTICS

(See foot notes 1 and 2 on timing diagrams)

Characteristic		Symbol	Min	Typ	Max	Unit
Cycle time	(1)	t_{CYC}	320		—	ns
Pulse width E low level	(2)	t_{WEL}	180		—	ns
Pulse width E high level	(3)	t_{WEH}	100		—	ns
Clock rise and fall time	(4)	t_r, t_f	—	—	20	ns
Control signal hold time	(5)	t_{HCE}	10	—	—	ns
Control signal set-up time	(6)	t_{SCE}	40	—	—	ns
Input data set-up time	(7)	t_{SDI}	120	—	—	ns
Input data hold time	(8)	t_{HDI}	10	—	—	ns
Output data set-up time (1 TTL load and CL = 50 pF)	(9)	t_{SDO}	—	—	150	ns
Output high impedance delay time (1 TTL load and CL = 50 pF)	(10)	t_{dz}	—	—	80	ns

RECEPTION CHARACTERISTICS

PERFORMANCE OF THE WHOLE RECEPTION CHAIN (input RAI or LEI, output RR1)

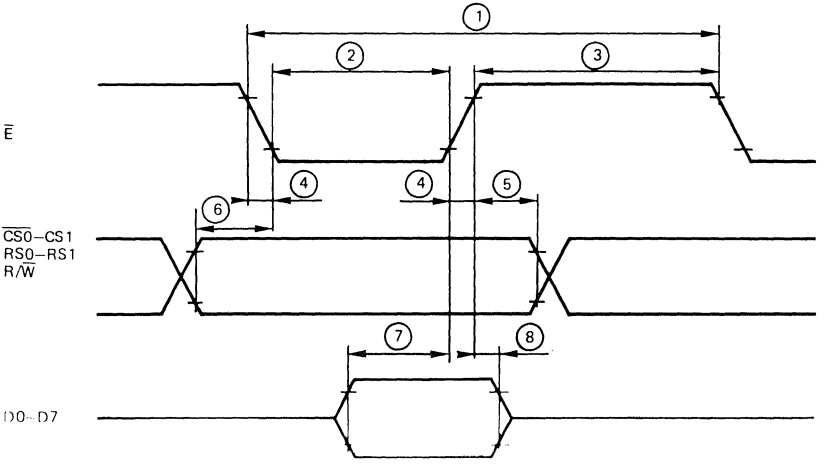
Characteristic	Symbol	Min	Typ	Max	Unit
Gain. (AGC gain = 0 dB, RxCCLK = 9600 Hz, V_{in}=775mV_{eff}, f = 2000 Hz)	G	— 0.5	—	0.5	dB
Total harmonic distortion (AGC gain = 0 dB, RxCCLK = 9600 Hz, V_{in}=775mV_{eff}, f = 2000 Hz)	TD	—	—	−58	dB
Equivalent RMS noise (AGC gain = 0 dB, RAI, LEI, EEI tied to AGND)	N		-	1.2	mV_{eff}

Note: Noise depends on AGC gain value.

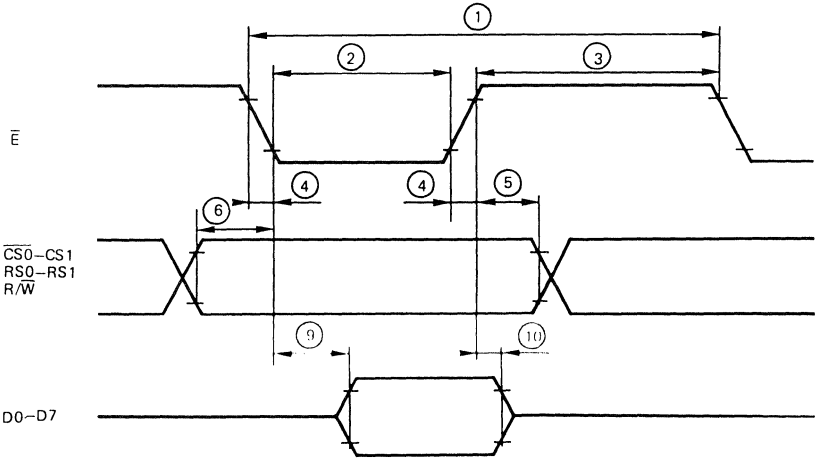
PERFORMANCE OF THE RECEPTION SUB-CHAIN (from RAI input to S/H2 input)

Parameter	Symbol	Min	Typ	Max	Unit
Total distortion (RxCCLK = 9600 Hz, V_{in} = 1.6 V_{eff}, f = 2000 Hz)	TD	—	—	— 72	dB

WRITE OPERATION



READ OPERATION



Notes:

1. Voltage levels shown are $V_{IL} < 0.4$ V, $V_{IH} > 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.2 V, unless otherwise specified.

RECEIVE BAND-PASS FILTER AND REJECTION FILTER (input RAI, output RFO)

Characteristic	Symbol	Min	Typ	Max	Unit
Low-Pass filter (Fs=288kHz)					
Reference gain (V_{in} = 775 mV_{eff}, f = 1800 Hz)	G_{ref}	— 0.5	—	0.5	dB
Relative gain to G_{ref}	G_{rel}				
0 Hz < f < 3000 Hz		— 0.4	—	0.3	dB
f = 3200 Hz		— 3	—	0.3	dB
f > 6250 Hz		—	—	— 60	dB
Group propagation delay time (f = 1800 Hz)	T_{gp}	—	—	300	µs
Group propagation delay time distortion (600 Hz < f < 3000 Hz)	T_{gpd}	—	—	360	µs
High-Pass filter (Fs=72kHz)					
Reference gain (V_{in} = 775 mV_{eff}, f 1800 Hz)	G_{ref}	— 0.5	—	0.5	dB
Relative gain to G_{ref}	G_{rel}				
0 Hz f 3000 Hz		0.4	—	0.3	dB
f 500 Hz		3	—	0.5	dB
f 100 Hz			—	— 25	dB
Group propagation delay time (f 1800 Hz)	T_{gp}			50	µs
Group propagation delay time distortion (600 Hz < f 3000 Hz)	T_{gpd}	—	—	450	µs
High-Pass filter and rejection filter (Fs=72kHz)					
Reference gain (V_{in} = 775 mV_{eff}, f = 1800 Hz)	G_{ref}	— 1	—	0	dB
Relative gain to G_{ref}	G_{rel}				
f = 100 Hz		—	—	— 25	dB
f = 370 Hz		—	—	— 27	dB
390 Hz < f < 450 Hz		—	—	— 30	dB
f = 470 Hz		—	—	— 27	dB
f = 900 Hz		—	—	0	dB
Group propagation delay time (f 1800 Hz)	T_{gp}	—	—	75	µs
Group propagation delay time distortion (600 Hz < f < 3000 Hz)	T_{gpd}	—	—	1400	µs

Note: The measurement frequencies are integer sub-multiples of filters sampling frequencies.

RECONSTRUCTION FILTER

Characteristic	Symbol	Min	Typ	Max	Unit
Reconstruction filter (Fs 288kHz)					
Reference gain (V_{in} = 775 mV_{eff}, f = 2000 Hz)	G_{ref}	— 0.3	—	0.3	dB
Relative gain to G_{ref} 0 Hz < f < 2900 Hz f = 3100 Hz f > 6000 Hz	G_{rel}	 — 0.4 — 3 —	 — — —	 0.3 0.3 — 60	 dB dB dB
Group propagation delay time (f = 1800 Hz)	T_{gp}	—	—	300	μs
Group propagation delay time distortion (600 Hz < f < 3000 Hz)	T_{gpd}	—	—	440	μs
Whole reception filtering chain (input RAI or LEI, output RFO)					
Reference gain (V_{in} = 775 mV_{eff}, f = 2000 Hz, RC3 = $A0)	G_{ref}	— 0.5	—	0.5	dB
Noise on RFO (RAI, LEI, EEI tied to AGND 250 Hz < f < 3200 Hz)	N_{rfo}	—	—	300	μV_{eff}

PERFORMANCE OF RESIDUAL SIGNAL CHANNEL AND A/D CONVERTER (input EEI, output RR2)

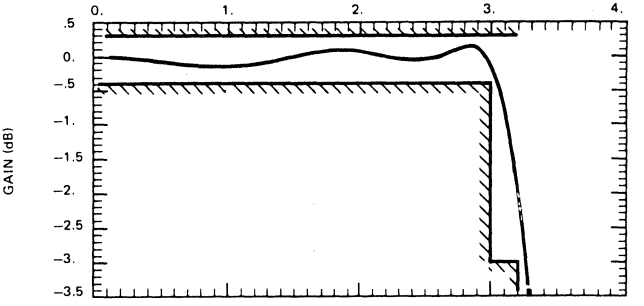
Characteristic	Symbol	Min	Typ	Max	Unit
Input voltage (peak to peak)	V_{in}	—	—	5	V
A/D converter resolution	R_{esh}	—	—	12	Bit
Analog increment	LSB	—	1.2	—	mV
Integral linearity error	E_{il}	— 16	—	16	LSB
Differential linearity error	E_{dl}	— 0.7	—	0.7	LSB
Offset voltage	V_{os}	— 100	—	100	LSB

AGC AMPLIFIER AND A/D CONVERTER (input AGC1, output RR1)

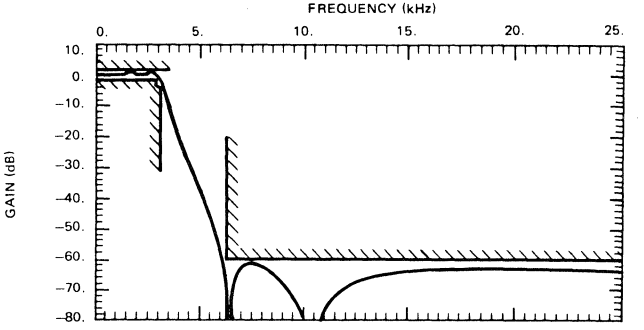
Characteristic	Symbol	Min	Typ	Max	Unit
Relative gain to programmed gain 0 dB ≤ AGC ≤ 24 dB 25.5 dB ≤ AGC ≤ 46.5 dB	G_{rel}	 — 0.5 — 1	 — —	 0.5 1	 dB dB
Offset voltage	V_{os}	— 70	—	70	LSB

CARRIER LEVEL DETECTOR (input AGC1, output CDR)

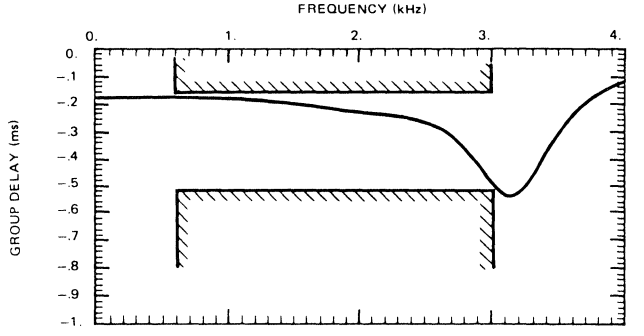
Characteristic	Symbol	Min	Typ	Max	Unit
Relative threshold to programmed gain 0 dB < AGC < 24 dB 25.5 dB ≤ AGC ≤ 46.5 dB	T_{rel}	 — 0.5 — 1	 — —	 0.5 1	 dB dB
Hysteresis	H_{yst}	2	—	3	dB
Input offset voltage 1st threshold pair 2nd threshold pair 3rd threshold pair	V_{os}	 — 1 — 2 — 3	 — — —	 1 2 3	 mV mV mV
Detection delay time 0 mV_{eff} to 775 mV_{eff} transition or 775 mV_{eff} to 0 V_{eff} transition	T_{dd}	1	—	3	ms



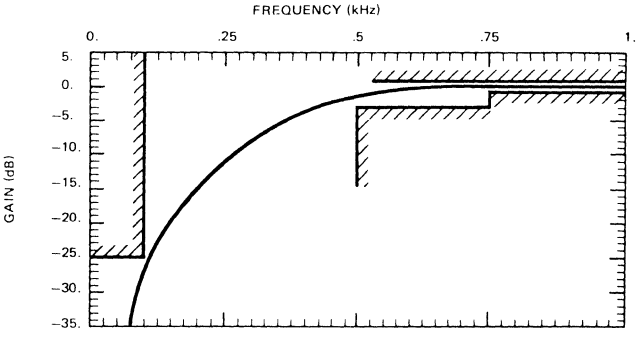
Rx LOW-PASS FILTER TYPICAL RESPONSE AND LIMITS CHART (Fs=288kHz)



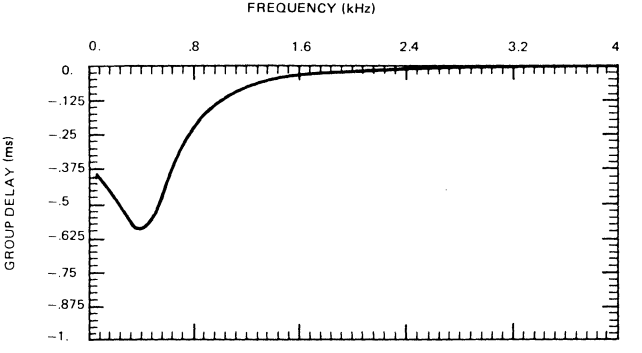
Rx LOW-PASS FILTER TYPICAL RESPONSE AND LIMITS CHART (Fs=288kHz)



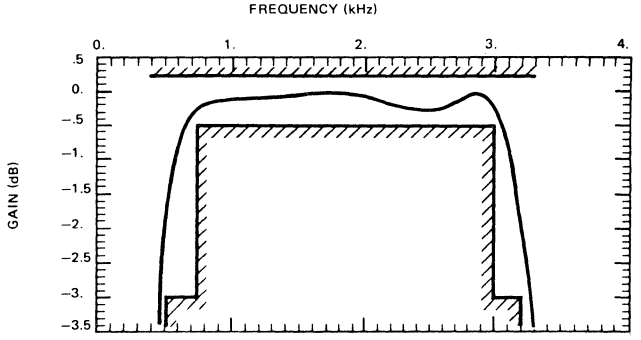
Rx LOW-PASS FILTER TYPICAL GROUP DELAY TIME AND LIMITS CHART (Fs=288kHz)



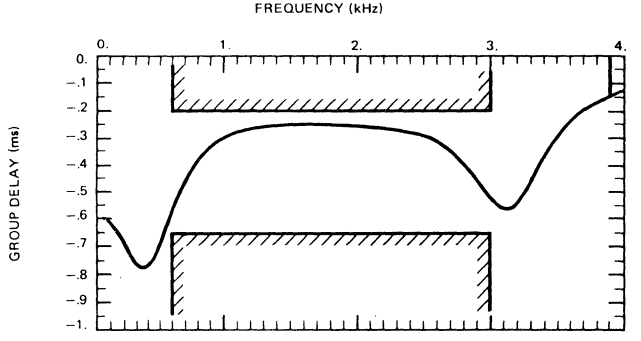
Rx HIGH-PASS FILTER TYPICAL RESPONSE AND LIMITS CHART (F_s = 72kHz)



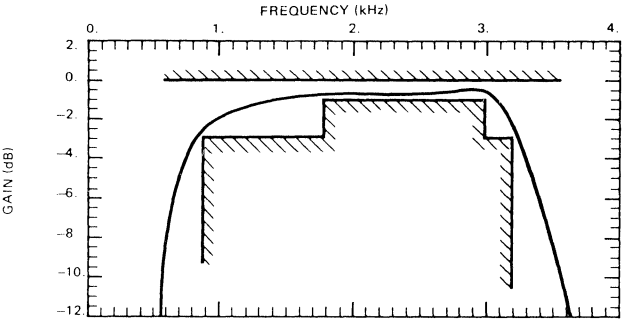
Rx HIGH-PASS FILTER TYPICAL GROUP DELAY TIME AND LIMITS CHART (F_s = 72kHz)



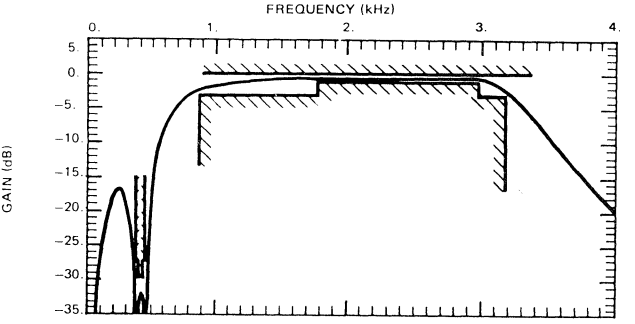
Rx BAND-PASS FILTER TYPICAL RESPONSE AND LIMITS CHART
(HP: Fs=72kHz, LP. Fs=288kHz)



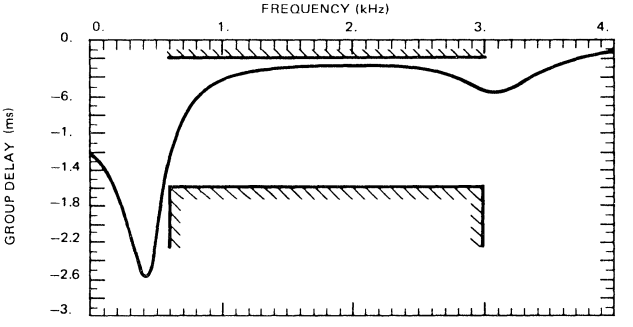
Rx BAND-PASS FILTER TYPICAL GROUP DELAY TIME AND LIMITS CHART
(HP: Fs=72kHz, LP: Fs=288kHz)



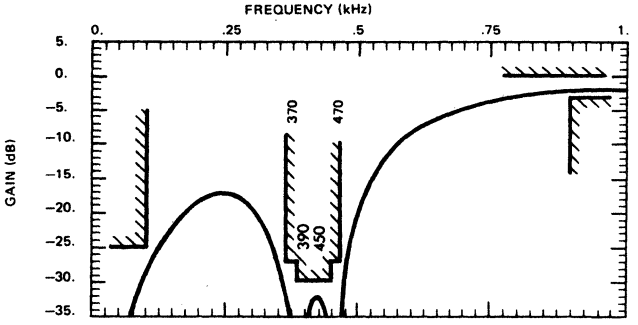
Rx BAND-PASS AND REJECTION FILTER TYPICAL RESPONSE AND LIMITS CHART
(HP AND REJ. : Fs=72kHz, LP: Fs=288kHz)



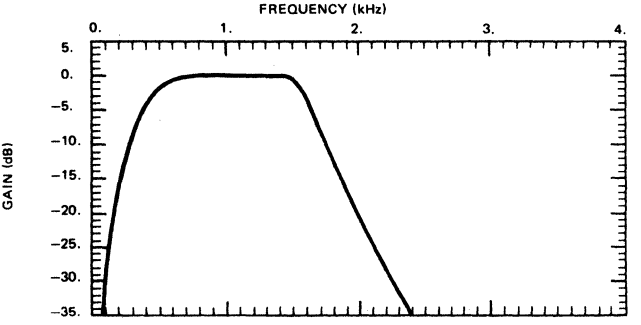
Rx BAND-PASS AND REJECTION FILTER TYPICAL RESPONSE AND LIMITS CHART
(HP AND REJ. :Fs=72kHz, LP: Fs=288kHz)



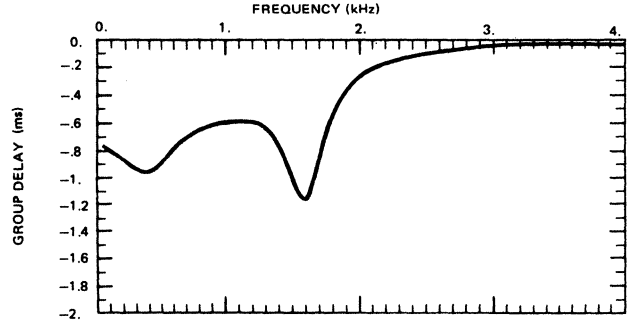
Rx BAND-PASS AND REJECTION FILTER TYPICAL GROUP DELAY TIME AND LIMITS CHART
(HP AND REJ. : Fs=72kHz, LP: Fs=288kHz)



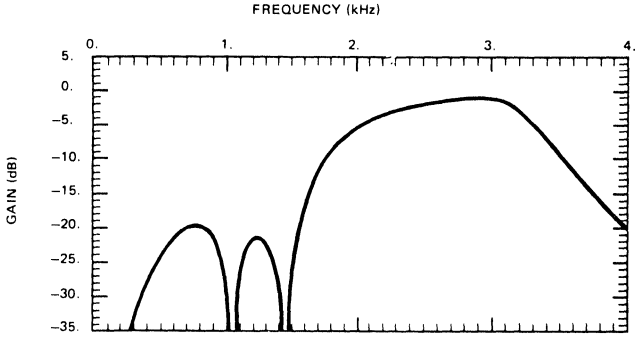
Rx HIGH-PASS AND REJECTION FILTER TYPICAL RESPONSE AND LIMITS CHART (F_s=72kHz



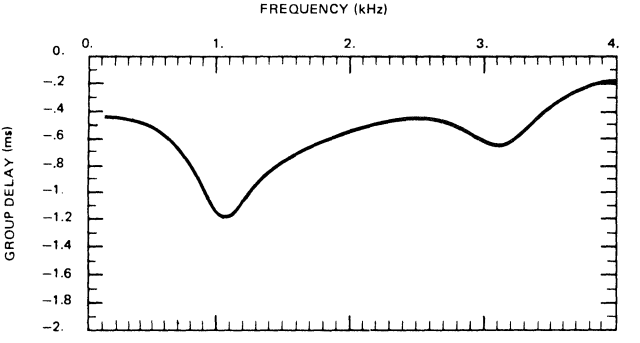
Rx BAND-PASS FILTER TYPICAL RESPONSE FOR V.22 MODE (LOW CHANNEL)
(HP: F_s=72kHz, LP: f_s=144kHz)



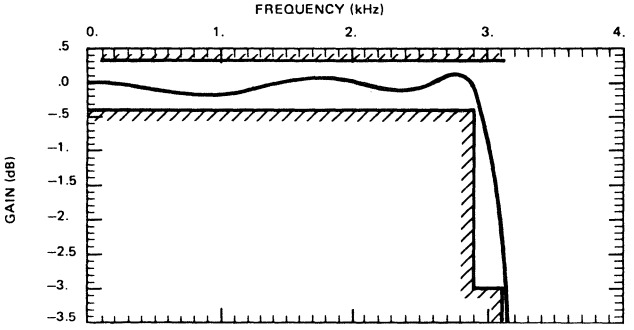
Rx BAND-PASS FILTER TYPICAL GROUP DELAY TIME FOR V.22 MODE (LOW CHANNEL)
(HP: Fs=72kHz, LP: Fs=144kHz)



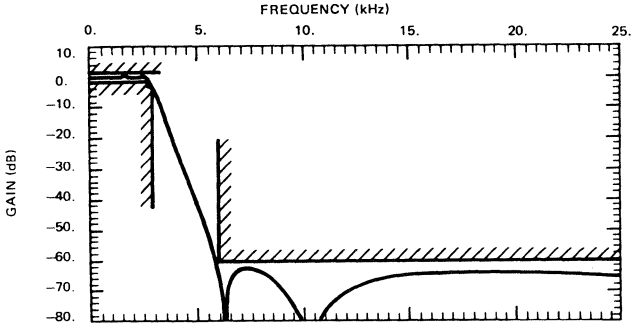
Rx BAND-PASS FILTER TYPICAL RESPONSE FOR V.22 MODE (HIGH CHANNEL)
(HP AND REJ.: Fs=144kHz, LP: Fs=288kHz)



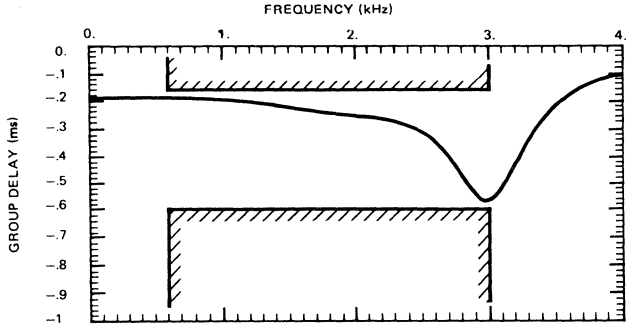
Rx BAND-PASS FILTER TYPICAL GROUP DELAY TIME FOR V.22 MODE (HIGH CHANNEL)
(HP AND REJ.: Fs=144kHz, LP: Fs=288kHz)



RECONSTRUCTION FILTER TYPICAL RESPONSE AND LIMITS CHART



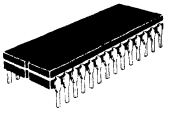
RECONSTRUCTION FILTER TYPICAL RESPONSE AND LIMITS CHARTS



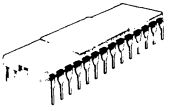
RECONSTRUCTION FILTER TYPICAL GROUP DELAY TIME AND LIMITS CHART

PHYSICAL DIMENSIONS

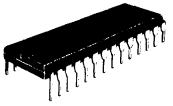
CB-132



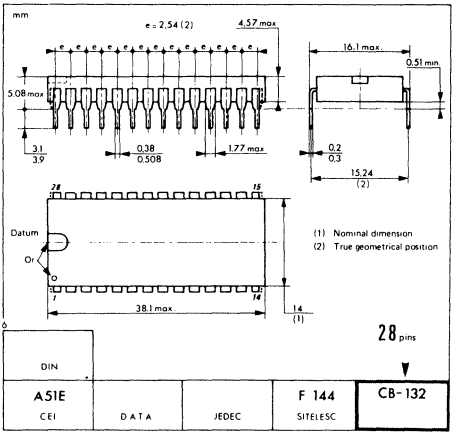
J SUFFIX
CERDIP PACKAGE



ALSO AVAILABLE
C SUFFIX
CERAMIC PACKAGE



P SUFFIX
PLASTIC PACKAGE



THOMSON COMPONENTS MOSTEK

TS68952

MODEM TRANSMIT/RECEIVE CLOCK GENERATOR

COMMUNICATIONS PRODUCTS

DATA SHEET

SILICON GATE CMOS

MODEM TRANSMIT/RECEIVE CLOCK GENERATOR

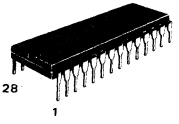
The TS68952 generates all the clock frequencies needed to implement standard voice-grade MODEMS up to 19200 bps according to the CCITT V.22, V.26, V.27, V.29, V.32 and V.33 or BELL 212A, 208 and 209 recommendations.

It can be associated with the TS68950 and the TS68951 to give a MODEM Analog Front-End Chip Set.

Main features:

- Independent Tx and Rx clock generators with Digital Phase Locked Loops (DPLLs).
- Tx DPLL synchronization on external terminal clock or internal Rx clock.
- Four external clocks available (plesiochronous on Tx and Rx channels): bit rate clock, baud rate clock, sampling clock and multiplexing clock.
- Chip programming and control via a standard 8 bit bus.

CASE CB-132

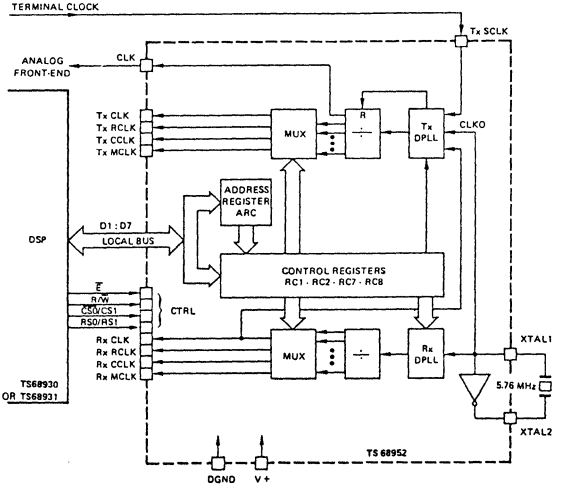


P SUFFIX
PLASTIC PACKAGE

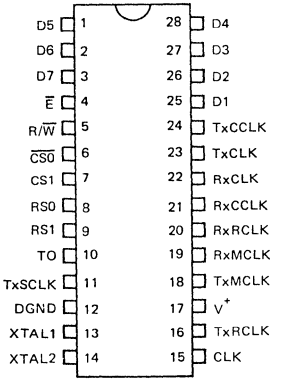
ALSO AVAILABLE
C SUFFIX
CERAMIC PACKAGE

J SUFFIX
CERDIP PACKAGE

BLOCK DIAGRAM



PIN ASSIGNMENT



PIN DESCRIPTION

Name	No.	Description
D5-D7	1-3	Data bus inputs to internal registers
$\overline{E}$	4	Enable input. Data are strobed on the positive transitions of this input
R/$\overline{W}$	5	Read/Write selection input. Internal registers can be written when R/W = 0. Reading mode is only used for Rx analog front-end chip
$\overline{CS0}$-CS1	6-7	Chip Select inputs. The chip set is selected when $\overline{CS0}$ = 0 and CS1 = 1
RS0-RS1	8-9	Register Select inputs. Used to select address or control registers
TO	10	Test Output. Must be left open
TxSCLK	11	Transmit Synchronizing Clock input. Normally tied to an external terminal clock. When this pin is tied to a permanent logical level, transmit DPLL free-runs or can be synchronized to the receive clock system.
DGND	12	Digital ground = 0 V. All digital signals are referenced to this pin
XTAL1	13	Crystal oscillator or pulse generator input
XTAL2	14	Crystal oscillator output
CLK	15	1.44 MHz Clock output. Useful for Tx and Rx analog front-end chips
TxRCLK	16	Transmit baud Rate Clock output
V^+	17	Positive power supply voltage = + 5 V ± 5 %
TxMCLK	18	Transmit Multiplexing Clock output
RxMCLK	19	Receive Multiplexing Clock output
RxRCLK	20	Receive baud Rate Clock output
RxCCLK	21	Receive Conversion Clock output
RxCLK	22	Receive bit rate Clock output
TxCLK	23	Transmit bit rate Clock output
TxCCLK	24	Transmit Conversion Clock output
D1-D4	25-28	Data bus inputs to internal registers (D0 is not used)

FUNCTIONAL DESCRIPTION

The TS68952 is a purely digital circuit that synthesises all the frequencies requested to implement synchronous voice-grade MODEMs from 1200 bps to 19200 bps. It consists of two clock generators using Digital Phase Locked Loops (DPLLs). Frequency programming and DPLL updating can be obtained through four control registers accessed by indirect or cyclical addressing.

This circuit is a part of a three chip Modem Analog Front-End that also includes the TS68950 transmitting analog interface and the TS68951 receiving analog interface.

POWER-UP INITIAL CONDITIONS

Following power-up, the eight transmit and receive clock outputs are undefined and may deliver any frequencies. Control registers RC1 and RC2 must be properly programmed to obtain the wanted operation.

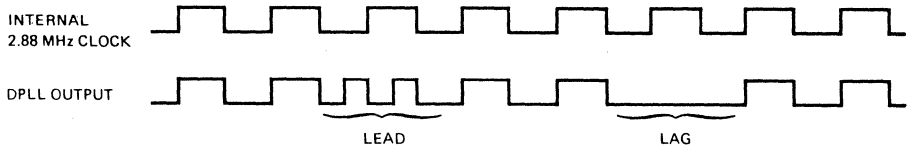


FIGURE 1 – DPLL LEAD AND LAG

CLOCK GENERATION

Master clock is obtained from either a crystal tied between XTAL1 and XTAL2 pins or an external signal connected to the XTAL1 pin; in this case, the XTAL2 pin should be left open. Clock frequency nominal value is 5.76 MHz, but 5.12 MHz and 5.40 MHz frequencies are also specified for particular applications.
The different transmit (Tx) and receive (Rx) clocks are obtained by frequency division in several counters and output selection through digital multiplexers. They can be synchronized on external signal via two independent digital phase locked loops (DPLL).

TRANSMIT DPLL

As shown on figure 1, the TxDPLL operates by adding or subtracting pulses to a 2.88 MHz internal clock, with a reference frequency that is a submultiple of the programmed "rate clock" frequency. This corresponds to phase leads or phase lags of about 350 ns duration, more precisely, two master clock periods.
The Tx DPLL can be synchronized on an external terminal clock tied to TxSCLK pin or on the receive bit clock RxCLK internally generated from the RxDPLL. It can also free-run without any phase shift, when the TxSCLK input is tied to a fixed logical level.

TRANSMIT CLOCKS

The TS68952 delivers four synchronous Tx clocks:

- a bit clock, TxCLK, whose frequency equals the bit rate of the MODEM,
- a baud clock, TxRCLK, whose frequency equals the baud rate of the MODEM,
- a conversion clock, TxCCLK, that gives the sampling frequency of the Tx converter,
- a multiplexing clock, TxMCLK, usable when several terminals are multiplexed on a single physical link.

The frequencies of these four clocks are programmable through RC1 and RC2 control registers. Their cyclical ratio is exactly 1 : 2, except for the 16.8 kHz frequency whose cyclical ratio is slightly modulated around 1 : 2, and their relative phase locking is ensured without user intervention, by periodic reset of the counters.

Immediate phasing of these clocks on the synchronizing external TxSCLK or internal RxCLK clock can be obtained through bit 7 of RC8 register. The content of this register is automatically cleared after phasing completion.

The TS68952 also delivers, on pin CLK, a 1.44 MHz clock that is synchronous with the Tx clock system and will be used as the main clock of the TS68950/51 analog interface circuits.

RECEIVE DPLL

RxDPLL phase shifts are performed by addition and subtraction of pulses from an internal 1.44 MHz clock under the control of RC8 register. Two modes of operation are provided:

- a coarse phase lag whose amplitude has been loaded into RC7 register, can be controlled by one bit of RC8 register. This mode is useful for a fast synchronization of the RxDPLL. The phase lag is obtained by suppressing a variable number of pulses at the input of the counters,
- a fine phase shift with lead or lag amplitude equal to two master clock periods, can be controlled by two bits of RC8. This mode corresponds to normal operation. The phase shifts are obtained by addition or suppression of pulses as indicated in figure 1.

RC8 register is automatically cleared when the programmed phase shift is completed. Simultaneous programming of Tx and Rx control bits of this register has to be avoided.

RECEIVE CLOCKS

The TS68952 delivers four Rx clocks with the same nominal frequency values as their Tx counterparts:

- a bit clock RxCLK,
- a baud clock RxRCLK,
- a conversion clock RxCCLK,
- a multiplexing clock RxMCLK.

The Rx and Tx output clocks are plesiochronous.

BIT CLOCK FREQUENCY PROGRAMMING (Tx AND Rx)

RC1 REGISTER							OUTPUT FREQUENCY (kHz)		
D7	D6	D5	D4	D3	D2	D1			
HB4	HB3	HB2	HB1	HR3	HR2	HR1	F_Q = 5.76 MHz	F_Q = 5.40 MHz	F_Q = 5.12 MHz
0	0	0	0				19.2		
0	0	0	1				16.8		
0	0	1	0				14.4		
0	0	1	1				12.0		
0	1	0	0				9.6		
0	1	0	1				7.2		6.4
0	1	1	0				6.4		
0	1	1	1				6.0		
1	0	0	0				4.8		
1	0	0	1				3.2	3.0	
1	0	1	0				2.4		
1	0	1	1				1.2		
1	1	0	0				0.6		
1	1	0	1				0.6		
1	1	1	0				0.6		
1	1	1	1				0.6		

F_Q = crystal oscillator frequency

RATE CLOCK FREQUENCY PROGRAMMING (Tx AND Rx)

RC1 REGISTER							OUTPUT FREQUENCY (kHz)		
D7	D6	D5	D4	D3	D2	D1			
HB4	HB3	HB2	HB1	HR3	HR2	HR1	F_Q = 5.76 MHz	F_Q = 5.40 MHz	F_Q = 5.12 MHz
				0	0	0	2.4		2.133
				0	0	1	2.0*		
				0	1	0	1.6**	1.5	
				0	1	1	1.2		
				1	0	0	0.6		
				1	0	1	0.6		
				1	1	0	0.6		
				1	1	1	0.6		

Note: Phase shift frequency of Tx DPLL is 600 Hz except for (*) 1000 Hz and for (**) 2000 Hz.

CONVERSION CLOCK FREQUENCY PROGRAMMING (Tx AND Rx)

RC2 REGISTER							OUTPUT FREQUENCY (kHz)		
D7	D6	D5	D4	D3	D2	D1			
HM3	HM2	HM1	HS2	HS1	HTHR	—	F_Q = 5.76 MHz	F_Q = 5.40 MHz	F_Q = 5.12 MHz
			0	0			9.6	9.0	8.533
			0	1			8.0	7.5	
			1	0			7.2		
			1	1			7.2		

MULTIPLEXING CLOCK FREQUENCY PROGRAMMING (Tx AND Rx)

RC2 REGISTER							OUTPUT FREQUENCY (kHz)
D7	D6	D5	D4	D3	D2	D1	
HM3	HM2	HM1	HS2	HS1	HTHR	—	F_Q = 5.76 MHz
0	0	0					1440
0	0	1					288
0	1	0					12
0	1	1					9.6
1	0	0					7.2
1	0	1					4.8
1	1	0					2.4
1	1	1					1.2

Tx SYNCHRONIZATION SIGNAL PROGRAMMING

RC2 REGISTER							SYNCHRONIZATION SIGNAL
D7	D6	D5	D4	D3	D2	D1	
HM3	HM2	HM1	HS2	HS1	HTHR	—	
					0		RxCLK
					1		TxSCLK (note 1)

Note: 1 - TxDPLL free-runs if there is no transition on this input.

Tx CLOCK GENERAL RESET

RC8 REGISTER (notes 2,3)							RESETTING TRANSITION
D7	D6	D5	D4	D3	D2	D1	
MPE	SPR	AVRE	VAL	INIT	—	—	
1	0	0	0	0			Next negative-going transition on synchronization clock

Note: 2 - RC8 register is cleared after the programmed control operation is completed.

Note: 3 - INIT bit is only used for test purpose.

Rx CLOCK PHASE SHIFT PROGRAMMING

RC8 REGISTER (note 2)							
D7	D6	D5	D4	D3	D2	D1	
MPE	SPR	AVRE	VAL	INIT	—	—	ACTION ON Rx DPLL
0	1	0	0	0			Phase lag of programmed amplitude
0	0	0	1	0			Phase lag of two main clock periods
0	0	1	1	0			Phase lead of two main clock periods

Rx CLOCK PHASE SHIFT AMPLITUDE PROGRAMMING

RC7 REGISTER							PHASE SHIFT IN DEGREES		
D7	D6	D5	D4	D3	D2	D1			
SP5	SP4	SP3	SP2	SP1	—	—	1200 bauds*	1600 bauds	NUMBER OF MASTER CLOCK PULSES SUPPRESSED
0	0	0	0	0			1.5	2	20
0	0	0	0	1			3	4	40
0	0	0	1	0			4.5	6	60
0	0	0	1	1			6	8	80
0	0	1	0	0			7.5	10	100
0	0	1	0	1			9	12	120
0	0	1	1	0			10.5	14	140
0	0	1	1	1			12	16	160
0	1	0	0	0			13.5	18	180
0	1	0	0	1			15	20	200
0	1	0	1	0			16.5	22	220
0	1	0	1	1			18	24	240
0	1	1	0	0			19.5	26	260
0	1	1	0	1			21	28	280
0	1	1	1	0			22.5	30	300
0	1	1	1	1			24	32	320
1	0	0	0	0			22.5	30	300
1	0	0	0	1			45	60	600
1	0	0	1	0			67.5	90	900
1	0	0	1	1			90	120	1200
1	0	1	0	0			112.5	150	1500
1	0	1	0	1			135	180	1800
1	0	1	1	0			157.5	210	2100
1	0	1	1	1			180	240	2400
1	1	0	0	0			202.5	270	2700
1	1	0	0	1			225	300	3000
1	1	0	1	0			247.5	330	3300
1	1	0	1	1			270	360	3600
1	1	1	0	0			292.5		3900
1	1	1	0	1			315		4200
1	1	1	1	0			337.5		4500
1	1	1	1	1			360		4800

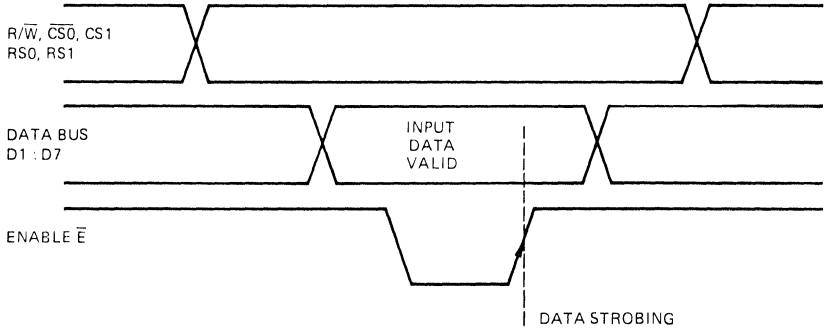
(*) 2400 bauds: multiply by two. 600 bauds: divide by two.

DATA BUS CONTROL

Six signals control the access from the bus to the internal registers according to the table and the timing diagram given below. Control registers are written using an indirect addressing mode where the internal address is stored in the 3 bit ARC register. After each write operation to a control register, the ARC register value is automatically increased by one. This allows cyclical addressing of the eight registers of the MODEM chip set.

R/$\overline{W}$	$\overline{CS0}$	$\overline{CS1}$	RS0	RS1	$\overline{E}$	ACCESSED REGISTER
0	0	1	1	0	↓	Address register ARC
0	0	1	1	1	↓	Control register whose address is in ARC

BUS TIMING DIAGRAM



DATA FORMAT

DATA LOADED IN ARC			
D7	D6	D5	
ARC3	ARC2	ARC1	ADDRESSED REGISTER
0	0	0	RC1
0	0	1	RC2
1	1	0	RC7
1	1	1	RC8

MAXIMUM RATINGS

RATING	MIN.	MAX.
V^+ supply voltage to DGND ground	- 0.3 V	7 V
Voltage at any input or output	DGND - 0.3 V	V^+ + 0.3 V
Current at any output	- 20 mA	20 mA
Power dissipation		500 mW
Operating temperature range	0 °C	70 °C
Storage temperature range	- 65 °C	+ 150 °C

OPERATING RANGE

Ambient temperature	V^+	DGND
$0\ °C \leq t_{amb} \leq +\ 70\ °C$	+ 5.0 V ± 5 %	0 V

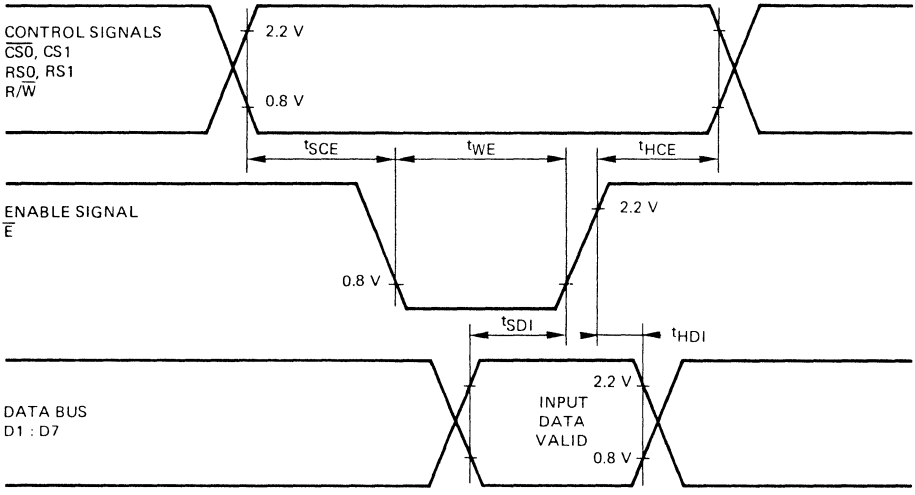
ELECTRICAL OPERATING CHARACTERISTICS

Unless otherwise noted, electrical characteristics are specified over the operating range. Typical values are given for $V^+ = 5.0$ V and $t_{amb} = 25$ °C.

CHARACTERISTIC	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Power dissipation						
Positive supply current	I^+				5.0	mA
Digital interface						
Input low level voltage	V_{IL}				0.8 V	V
Input high level voltage	V_{IH}		2.2			V
Input low level current	I_{IL}	$DGND \leq V_I \leq V_{IL\ max}$	- 10		10	μA
Input high level current	I_{IH}	$V_{IH\ min} \leq V_I \leq V^+$	- 10		10	μA
Output low level current	V_{OL}	$I_O = 2.5$ mA			0.4	V
Output high level current	V_{OH}	$I_O = -2.5$ mA	2.4			V
Crystal oscillator interface						
Input low level voltage	V_{IL}				1.5	V
Input high level voltage	V_{IH}		3.5			V
Input low level current	I_{IL}	$DGND \leq V_I \leq V_{IL\ max}$	- 15			μA
Input high level current	I_{IH}	$V_{IH\ min} \leq V_I \leq V^+$			15	μA

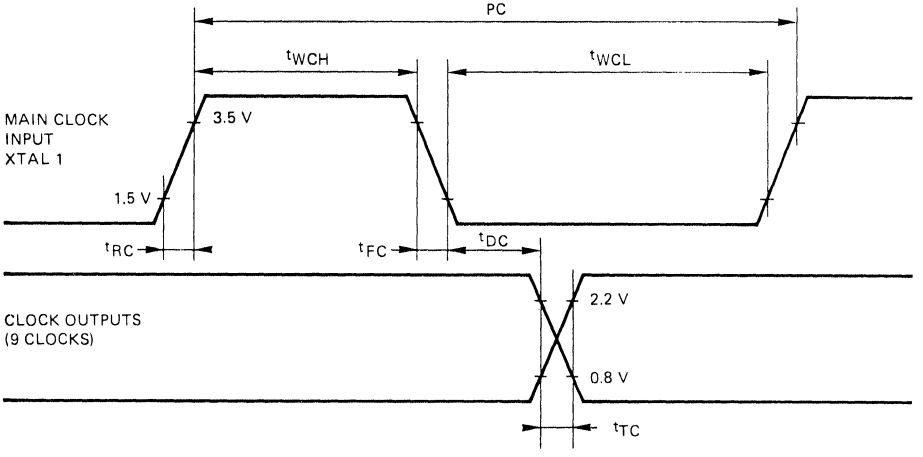
TIMING CHARACTERISTICS

CHARACTERISTIC	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Data bus access						
Control signals set-up time	t_{SCE}	$\overline{CS0}$, CS1, RS0, RS1, R/$\overline{W}$ to $\overline{E}$	40			ns
Control signals hold time	t_{HCE}	$\overline{CS0}$, CS1, RS0, RS1, R/$\overline{W}$ to $\overline{E}$	10			ns
Data-in set-up time	t_{SDI}	D1 : D7 to $\overline{E}$	120			ns
Data-in hold time	t_{HDI}	D1 : D7 to $\overline{E}$	10			ns
Enable signal low level width	t_{WE}	$\overline{E}$		180		ns

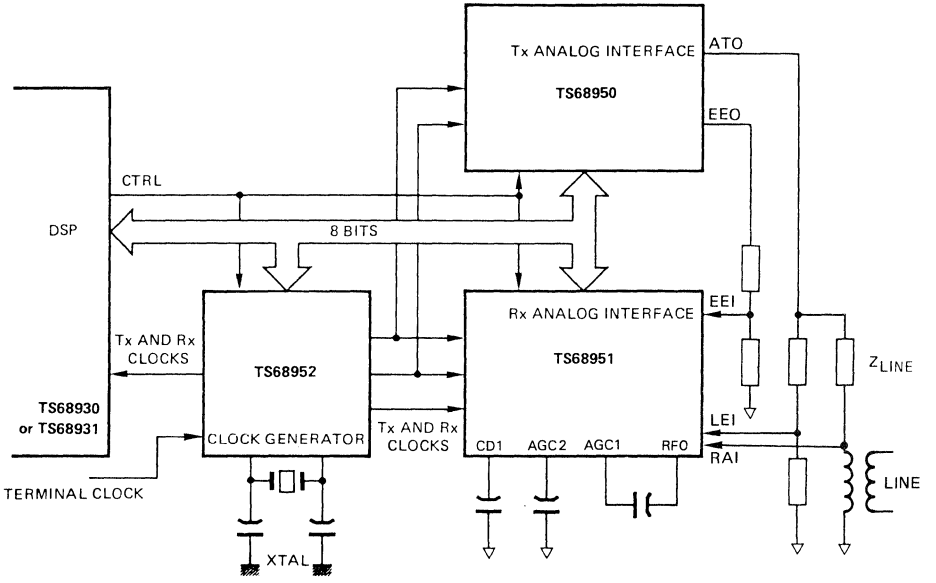


TIMING CHARACTERISTICS (continued)

CHARACTERISTIC	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Clock wave forms						
Main clock period	PC	XTAL1 input	150	173.6		ns
Main clock low level width	t_{WCL}	XTAL1 input	50			ns
Main clock high level width	t_{WCH}	XTAL1 input	50			ns
Main clock rise time	t_{RC}	XTAL1 input			50	ns
Main clock fall time	t_{FC}	XTAL1 input			50	ns
Clock output delay time	t_{DC}	All clock outputs CL = 50 pF			500	ns
Clock output transition time	t_{TC}	All clock outputs CL = 50 pF			100	ns



TYPICAL APPLICATION



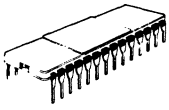
MODEM ANALOG FRONT-END CHIP SET

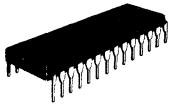
NOTE: Digital ground

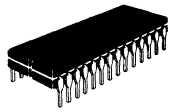
Analog ground

PHYSICAL DIMENSIONS

CB-132





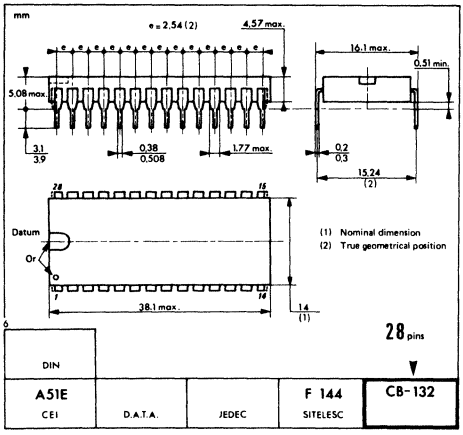


ALSO AVAILABLE

C SUFFIX
CERAMIC PACKAGE

P SUFFIX
PLASTIC PACKAGE

J SUFFIX
CERDIP PACKAGE



CHAPTER 8 - DATA COMMUNICATIONS ICs

THOMSON COMPONENTS
MOSTEK

MK68590 (P,N)

LOCAL AREA NETWORK CONTROLLER FOR ETHERNET

COMMUNICATIONS PRODUCTS

FEATURES

- ☐ 100% compatible with Ethernet and IEEE 802.3 specifications
- ☐ Data packets moved by block transfers over a processor bus (on-board DMA controller 24-bit linear address space)
- ☐ Buffer management
- ☐ Packet framing
- ☐ Preamble and Cyclic Redundancy Check (CRC) insertion
- ☐ Preamble stripping and CRC verification
- ☐ General 16-bit microprocessor bus interface compatible with popular processors (68000, 8086, Z8000, LSI-11)
- ☐ Cable fault detection
- ☐ Multicast logical address filtration
- ☐ Collision handling and retry
- ☐ Scaled N-channel MOS VLSI technology
- ☐ 48-pin DIP
- ☐ Single 5-volt power supply
- ☐ Single phase TTL level clock
- ☐ All inputs and outputs TTL compatible
- ☐ Completely compatible with companion Serial Interface Adapter (SIA) chip MK68591/2.

DESCRIPTION

The MK68590-LANCE™ (Local Area Network Controller for Ethernet) is a 48-pin VLSI device that simplifies the interfacing of a microcomputer or a minicomputer to an Ethernet Local Area Network. This chip operates in a local environment that includes a closely coupled memory and microprocessor. The LANCE uses scaled N-channel MOS technology and is compatible with several microprocessors.

LANCE is a trademark of Thomson Components - Mostek Corporation.

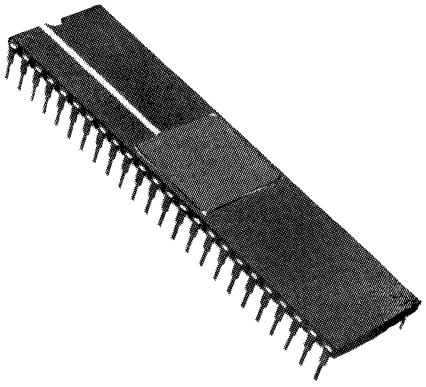


Figure 1. MK68590

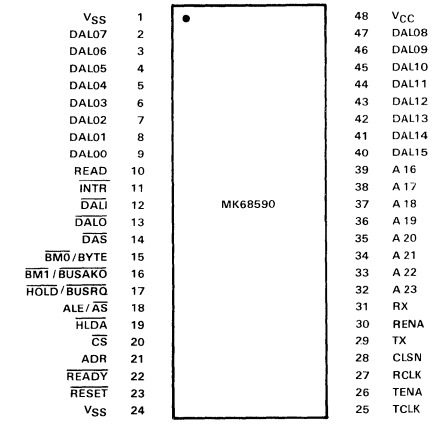


Figure 2. LANCE Pin Assignment

PIN DESCRIPTION

DAL00-DAL15
(Data/Address Bus)
Input/Output Three State. The time multiplexed Address/Data bus. These lines will be driven as a bus master and as a bus slave.

READ
Input/Output Three State. Indicates the type of operation to be performed in the current bus cycle. When it is a bus master, LANCE drives this signal.

LANCE as bus slave:
High - The chip places data on the DAL lines.
Low - The chip takes data off the DAL lines.
LANCE as bus master:
High - The chip takes data off the DAL lines.
Low - The chip places data on the DAL lines.

$\overline{\text{INTR}}$
(Interrupt)
Output Open Drain. When enabled, an attention signal that indicates the occurrence of one or more of the following events: a message reception or transmission has completed or an error has occurred during the transaction; the initialization procedure has completed; or a memory error has been encountered. Setting INEA in CSR0 (bit 06) enables $\overline{\text{INTR}}$.

$\overline{\text{DALI}}$
(Data/Address Line In)
Output Three State. An external bus transceiver control line. When LANCE is a bus master and reads from the DAL lines, $\overline{\text{DALI}}$ is asserted during the data portion of the transfer.

$\overline{\text{DALO}}$
(Data/Address Line Out)
Output Three State. An external bus transceiver control line. When LANCE is a bus master and drives the DAL lines, $\overline{\text{DALO}}$ is asserted during the address portion of a read transfer or for the duration of a write transfer.

$\overline{\text{DAS}}$
(Data/Strobe)
Input/Output Three State. Defines the data portion of the bus transaction. DAS is driven only as a bus master.

$\overline{\text{BM0}}$, $\overline{\text{BM1}}$ or BYTE, $\overline{\text{BUSAKO}}$
(Byte Mask)
Output Three State. Pins 15 and 16 are programmable through bit (00) of CSR3 (known as BCON). Asserting $\overline{\text{RESET}}$ clears CSR3.

f BCON = 0
PIN 16 = $\overline{\text{BM1}}$ (Output Three State)
PIN 15 = $\overline{\text{BM0}}$ (Output Three State)

BM0, BM1 Byte Mask. Indicates the byte(s) of a bus transaction to be read or written. The BM lines are ignored as a bus slave and assume word transfers only. The LANCE drives the BM lines only when it is a bus master. Byte selection occurs as follows:

$\overline{\text{BM1}}$	$\overline{\text{BM0}}$	
Low	Low	Whole Word
Low	High	Byte of DAL 08 - DAL 15
High	Low	Byte of DAL 00 - DAL 07
High	High	None

If BCON = 1
PIN 16 = $\overline{\text{BUSAKO}}$ (Output)
PIN 15 = BYTE (Output Three State)

BYTE. An alternate byte selection line. Byte selection occurs when the BYTE and DAL (00) lines are latched during the address portion of the bus transaction. BYTE, $\overline{\text{BM0}}$ and $\overline{\text{BM1}}$ are ignored when LANCE is a bus slave. There are two modes of ordering bytes depending on bit (02) of CSR3, (known as BSWP). This programmable ordering of upper and lower bytes when using BYTE and DAL (00) as selection signals is required to make the ordering compatible with various 16-bit microprocessors.

	BSWP = 0		BSWP = 1	
BYTE	DAL(00)	BYTE	DAL(00)	
Low	Low	Low	Low	Whole Word
Low	High	Low	High	Illegal Condition
High	High	High	Low	Upper Byte
High	Low	High	High	Lower Byte

$\overline{\text{BUSAKO}}$. The DMA daisy chain output.

$\overline{\text{HOLD}}$/$\overline{\text{BUSRQ}}$
(Bus Hold Request)
Input/Output Open Drain. LANCE asserts this signal when it requires access to memory. $\overline{\text{HOLD}}$ is held low for the entire bus transaction. This bit is programmable through bit (00) of CSR3 (known as BCON). In the daisy chain DMA mode (BCON = 1) $\overline{\text{BUSRQ}}$ is asserted only if $\overline{\text{BUSRQ}}$ is inactive prior to assertion. Bit (00) of CSR3 is cleared when $\overline{\text{RESET}}$ is asserted.

CSR3(00) BCON = 0
PIN 17 = $\overline{\text{HOLD}}$ (Output Open Drain)
CSR3(00) BCON = 1
PIN 17 = $\overline{\text{BUSRQ}}$ (Output Open Drain)
$\overline{\text{BUSRQ}}$ will be asserted only if PIN 17 is high prior to assertion.

ALE/$\overline{\text{AS}}$
(Address Latch Enable)
Output Three State. Used to demultiplex the DAL lines and define the address portion of the bus cycle. This

pin is programmable through bit (01) of CSR3. As ALE, the signal transitions from high to low at the end of the address portion of the bus the address portion of the bus transaction and remains low during the entire data portion of the transaction. As $\overline{AS}$, the signal transitions from low to high at the end of the address portion of the bus transaction and remains high throughout the entire data portion of the transaction. The LANCE drives the ALE/$\overline{AS}$ line only as a bus master.

CSR3(01) ACON = 0
PIN 31 = ALE
CSR3(01) ACON = 1
PIN 31 = $\overline{AS}$

$\overline{HLDA}$
(Bus Hold Acknowledge)
Input. A response to $\overline{HOLD}$ indicating that the LANCE is the Bus Master. $\overline{HLDA}$ stops its response when $\overline{HOLD}$ ends its assertion.

$\overline{CS}$
(Chip Select)
Input. When asserted, $\overline{CS}$ indicates LANCE is the slave device of the data transfer. $\overline{CS}$ must be valid throughout the data portion of the bus cycle.

ADR
(Register Address Port Select)
Input. When $\overline{CS}$ is asserted, ADR indicates which of the two register ports is selected. ADR must be valid throughout the data portion of the bus cycle.

ADR	PORT
Low	Register Data Port
High	Register Address Port

$\overline{READY}$
Input/Output Open Drain. When the LANCE is a bus master, $\overline{READY}$ is an asynchronous acknowledgement from external memory that will complete the data transfer. As a bus slave, the chip asserts $\overline{READY}$ when it has put data on the bus, or is about to take data off the bus. $\overline{READY}$ is a response to $\overline{DAS}$. $\overline{READY}$ negates after $\overline{DAS}$ negates. Note: If $\overline{DAS}$ or $\overline{CS}$ deassert prior to the assertion of $\overline{READY}$, $\overline{READY}$ cannot assert.

$\overline{RESET}$
Input. Bus reset signal. Causes LANCE to cease operation and to enter an idle state.

TLCK
(Transmit Clock)
Input. Normally a free-running 10 MHz clock (crystal-controlled within 0.01% accuracy).

TENA
(Transmit Enable)
Output. Transmit Output Stream Enable. A level asserted with the transmit output bit stream, TX, to enable the external transmit logic.

RCLK
(Receive Clock)
Input. Normally a 10 MHz square wave synchronized to the receive data and present only while receiving an input bit stream.

CLSN
(Collision)
Input. A logical input that indicates that a collision is occurring on the channel.

TX
(Transmit)
Output. Transmit output bit stream.

RENA
(Receive Enable)
Input. A logical input that indicates the presence of data on the channel.

RX
(Receive)
Input. Receive input bit stream.

A16-A23
(High-Order Address Bus)
Output Three State. The additional address bits necessary to extend the DAL lines to produce a 24-bit address. These lines will be driven only as a bus master.

VCC
Power supply pin. +5 VDC ±5 percent.

VSS
Ground. 0 VDC.

FUNCTIONAL CAPABILITIES

The LANCE interfaces to a microprocessor bus characterized by time-multiplexed address and data lines. Typically, data transfers are 16 bits wide but byte transfers occur if the buffer memory address boundaries are odd. The address bus is 24 bits wide.

The Ethernet packet format consists of 64-bit preamble, a 48-bit destination address, a 48-bit source address, a 16-bit type field, and from a 46 to 1500 byte data field terminated with a 32-bit CRC. The packets' variable widths accommodate both short-status command and terminal traffic packets and long data packets to printers and disks (1024-byte disk sectors, for example). Packets are spaced a minimum of 9.6 μsec apart to allow one node enough time to receive back-to-back packets.

The LANCE operates in a minimal configuration that requires close coupling between local memory and a processor. The local memory provides packet buffering

for the chip and serves as a communication link between chip and processor. During initialization, the control processor loads the starting address of the initialization block plus the operating mode of the chip via two ports that can access four control registers into LANCE. The host processor talks directly to LANCE only during this initial phase. All further communications are handled via a Direct Memory Access (DMA) machine under microword control contained within LANCE. Figure 3 shows a block diagram of the LANCE and SIA device used to create an Ethernet interface for a computer system.

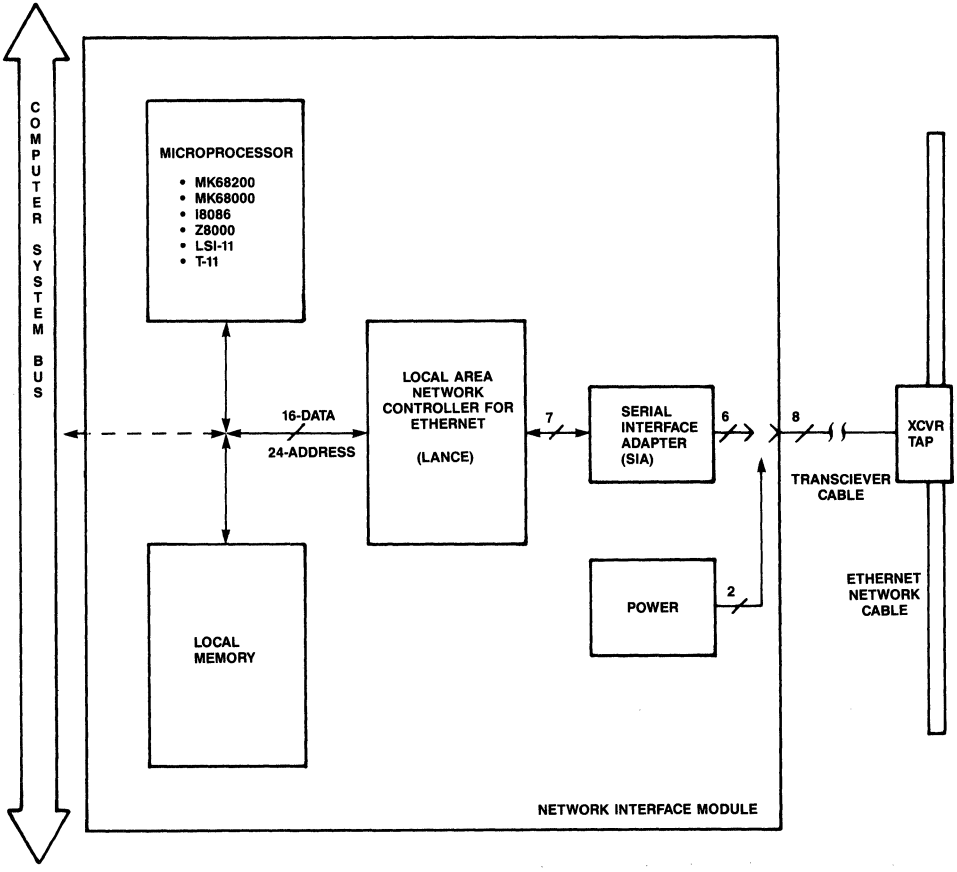


Figure 3. Ethernet Local Area Network System Block Diagram

FUNCTIONAL DESCRIPTION

SERIAL DATA HANDLING

LANCE provides the Ethernet interface as follows. In the transmit mode (since there is only one transmission path, Ethernet is a half duplex system), the LANCE reads data from a transmit buffer by using DMA and appends the preamble, sync pattern (two ones after alternating ones and zeros in the preamble), and calculates and appends the complement of the 32-bit CRC. In the receive mode, the destination address, source address, type, data, and CRC fields are transferred to memory via DMA cycles. The CRC is calculated as data and transmitted CRC is received. At the end of the packet, if this calculated CRC does not agree with a constant, an error bit is set in RDM1 of the receiver descriptor ring. In the receive mode, LANCE accepts packets under four modes of operation. The first mode is a full comparison of the 48-bit destination address in the packet with the node address that was programmed into the LANCE during an initialization cycle. There are two types of logical addresses. One is a group type mask where the 48-bit address in the packet is put through a hash filter in order to map the 48-bit physical addresses into 1 of 64 logical groups. This mode can be useful if simultaneously sending packets to all of one type of a device on the network. (i.e., send a packet to all file servers or all printer servers). The second logical address is a multicast address where all nodes on the network receive the packet. The last receive mode of operation is the so called "promiscuous mode" in which a node will accept all packets on the cable regardless of their destination address.

COLLISION DETECTION AND IMPLEMENTATION

The Ethernet CSMA/CD network access algorithm is implemented completely within LANCE. In addition to listening for a clear network cable before transmitting, Ethernet handles collisions in a predetermined way. Should two transmitters attempt to seize the network cable at the same time, they will collide, and the data on the network cable will be garbled. LANCE is constantly monitoring the Collision (CLSN) pin. This signal is generated by the transceiver when the signal level on the network cable indicates the presence of signals from two or more transmitters. If LANCE is transmitting when CLSN is asserted, it will continue to transmit the preamble (collisions normally occur while the preamble is being transmitted), then will "jam" the network for 32 bit times (3.2 microseconds). This jamming ensures that all nodes have enough time to detect the collision. The transmitting nodes then delay a random amount of time according to the "truncated binary backoff" algorithm defined in the Ethernet specification to minimize the probability of the colliding nodes having multiple collisions with each other. After 16 abortive attempts to transmit a packet, LANCE will report a RTRY error due to excessive collisions and step over the transmitter buffer. During reception, the detection of a collision causes that reception to be aborted. Depending on when the collision occurred, LANCE will treat this packet as an error packet if the packet has an address mismatch, as a runt packet (a packet that has less than 64 bytes), or as a legal length packet with a CRC error.

Fatal error reporting is provided by LANCE through a microprocessor interrupt and error flags in CSR0. Significant error conditions are late collision on received data, transmitter on longer than 1518 bytes, missed packet, and memory error (in which the memory did not respond, READY did not assert, to a memory cycle request).

Additional errors are reported through bits in the descriptor rings (on a buffer by buffer basis). Receive error conditions include framing, CRC and buffer errors, and overflow. Transmit descriptor rings have error bits indicating buffer, underflow, late collision, and loss of carrier. Additionally, transmit descriptor rings have a bit indicating that the transmitter has unsuccessfully tried to transmit over a busy communication link.

Transmit descriptor rings also have nine bits reserved for a Time Domain Reflectometry counter (TDR). On the occurrance of a collision, the value in the TDR will give the number of system clocks until the collision, which can be used to determine the distance to the fault.

BUFFER MANAGEMENT

A key feature of the LANCE and its DMA channel is the flexibility and speed of communication between the LANCE and the host microprocessor through common memory locations. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings, as shown in Figure 4. These rings control both transmit and receive operations. Up to 128 tasks may be queued on a descriptor ring for execution by the LANCE. Each entry in a descriptor ring holds a pointer to a data memory buffer and an entry for the data buffer length. Data buffers can be chained or cascaded to handle a long packet in multiple data buffer areas. The LANCE searches the descriptor rings to determine the next empty buffer. This enables it to chain buffers together or to handle back-to-back packets. As each buffer is filled, an "own" bit is reset, signaling the host processor to empty this buffer.

MICROPROCESSOR INTERFACE

The parallel interface of LANCE has been designed to be "friendly", or easy to interface, to many popular 16-bit microprocessors. These microprocessors include the MK68000, Z8000, 8086, LSI-11, T-11, and MK68200 (the MK68200 is a 16-bit single chip microcomputer produced by Mostek, the architecture of which is modeled after the MK68000). LANCE has a wide 24-bit linear address space when in the Buster Master Mode,

allowing it to DMA the entire address space of the above microprocessors. LANCE uses no segmentation or paging methods. As such, LANCE addressing is closest to MK68000 addressing, but is compatible with the other microprocessors. When LANCE is a bus master, a programmable mode of operation allows byte addressing, either by employing a Byte/Word control signal (much like that used on the 8086 or the Z8000) or by using an Upper Data Strobe/Lower Data Strobe much like that used on the MK68000, LSI-11 and MK68200 microprocessors. A programmable polarity on the Address Strobe signal eliminates the need for external logic. LANCE interfaces with multiplexed and demultiplexed data busses and features control signals for address/data bus transceivers.

After the initialization routine, packet reception or transmission, transmitter timeout error, a missed packet, or memory error, LANCE generates an interrupt to the host microprocessor.

The cause of the interrupt is ascertained by reading CSR0. Bit (06) of CSR0, INEA, enables or disables interrupts to the microprocessor. In a polling mode, BIT (07) of CSR0 is sampled to determine when an interrupt causing condition occurred.

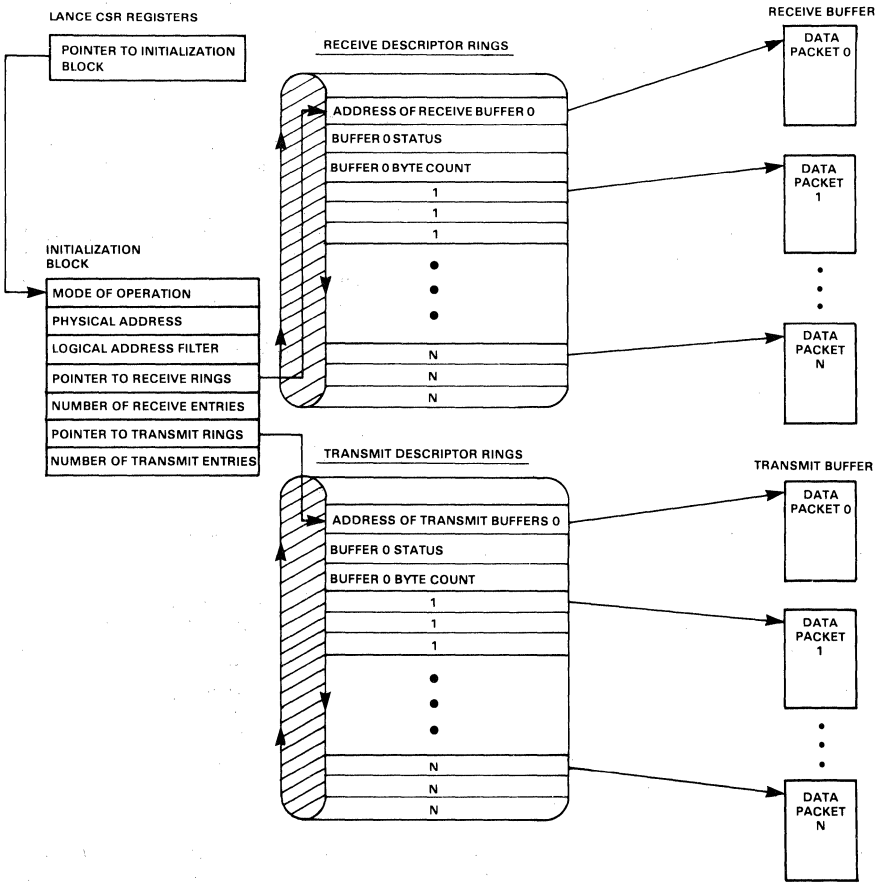


Figure 4. LANCE Memory Management

LANCE INTERFACE DESCRIPTION

ALE, $\overline{DAS}$ and $\overline{READY}$ time all data transfers from the LANCE in the Bus Master mode. The automatic adjustment of the LANCE cycle by the $\overline{READY}$ signal allows synchronization with variable cycle time memory due either to memory refresh or to dual port access. Bus cycles are a minimum of 600 ns long and can be increased in 100 ns increments.

READ SEQUENCE

At the beginning of a read cycle, valid addresses are placed on DAL00-DAL15 and A16-A21. The BYTE Mask signals ($\overline{BM0}$ and $\overline{BM1}$) become valid at the beginning of this cycle as does READ, indicating the type of cycle. The trailing edge of ALE or $\overline{AS}$ strobes the addresses A0-A15 into the external latches. Approximately 100 ns later, DAL00-DAL15 go into a three state mode. There is a 50 ns delay to allow for transceiver turnaround, then DAS falls low to signal the beginning of the data portion of the cycle. At this point in the cycle, the LANCE stalls waiting for the memory device to assert $\overline{READY}$. Upon assertion of $\overline{READY}$, $\overline{DAS}$ makes a transition from a zero to a one, latching memory data. ($\overline{DAS}$ is low for a minimum of 200 ns).

The bus transceiver controls, $\overline{DALI}$ and $\overline{DALO}$, control the bus transceivers. DALI signals to strobe data toward the LANCE and DALO signals to strobe data or addresses away from the LANCE. During a read cycle, $\overline{DALO}$ goes inactive before $\overline{DALI}$ goes active to avoid "spiking" of bus transceivers.

WRITE SEQUENCE

The write cycle begins exactly like a read cycle with the READ line remaining inactive. After ALE or $\overline{AS}$ pulse, the DAL00-DAL15 change from addresses to data. $\overline{DAS}$ goes active when the DAL00-DAL15 are stable. This data remains valid on the bus until the memory device asserts $\overline{READY}$. At this point, $\overline{DAS}$ goes inactive, latching data into the memory device. Data is held for 75 ns after the negation of $\overline{DAS}$.

LANCE INTERFACE DESCRIPTION — BUS SLAVE MODE

The LANCE enters the Bus Slave Mode whenever $\overline{CS}$ becomes active. This mode must be entered whenever writing or reading the four status control registers (CSR0, CSR1, CSR2, and CSR3) and the register address pointer (RAP). RAP and CSR0 may be read or written to at any time, but the LANCE must be stopped (CSR0 bit 02) when CSR1, CSR2, or CSR3 is to be written to or read.

MK68590 ELECTRICAL SPECIFICATION

ABSOLUTE MAXIMUM RATINGS

Temperature Under Bias . . . . . −25°C to +125°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin with Respect to Ground . . . . . −0.3 V to +7 V
Power Dissipation . . . . . 2.0 W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

T_A=0°C to 70°C, V_{CC} = +5 V ±5 percent unless otherwise specified.

SYMBOL	PARAMETER	MIN	MAX	UNITS
V_{IL}		−0.5	+0.8	V
V_{IH}		+2.0	V_{CC} +0.5	V
V_{OL}	@ I_{OL} = 3.2 mA		+0.5	V
V_{OH}	@ I_{OH} = −0.4 mA	+2.4		V
I_{IL}	@ V_{in} = 0.4 to V_{CC}		±10	μA

CAPACITANCE

F=1 MHz

SYMBOL	PARAMETER	MIN	MAX	UNITS
C_{IN}			10	pf
C_{OUT}			10	pf
C_{IO}			20	pf

AC TIMING SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = +5 V ±5 percent, unless otherwise specified.

NO.	SIGNAL	SYMBOL	PARAMETER	TEST CONDITIONS	MIN (ns)	TYP (ns)	MAX (ns)
1	TCLK	T_{TCT}	TCLK period		99		101
2	TCLK	T_{TCL}	TCLK low time		45		55
3	TCLK	T_{TCH}	TCLK high time		45		55
4	TCLK	T_{TCR}	Rise time of TCLK		0		8
5	TCLK	T_{TCF}	Fall time of TCLK		0		8
6	TENA	T_{TEP}	TENA propagation delay after the rising edge of TCLK	CL = 50 pf			95
7	TENA	T_{TEH}	TENA hold time after the rising edge of TCLK	CL = 50 pf	5		

AC TIMING SPECIFICATIONS (Continued)
T_A = 0°C to 70°C, V_{CC} = +5 V ±5 percent, unless otherwise specified.

NO.	SIGNAL	SYMBOL	PARAMETER	TEST CONDITIONS	MIN (ns)	TYP (ns)	MAX (ns)
8	TX	T_{TDP}	TX data propagation delay after the rising edge of TCLK	CL=50 pf			95
9	TX	T_{TDH}	TX data hold time after the rising edge of TCLK	CL=50 pf	5		
10	RCLK	T_{RCT}	RCLK period		85		118
11	RCLK	T_{RCH}	RCLK high time		38		
12	RCLK	T_{RCL}	RCLK low time		38		
13	RCLK	T_{RCR}	Rise time of RCLK		0		8
14	RCLK	T_{RCF}	Fall time of RCLK		0		8
15	RX	T_{RDR}	RX data rise time		0		8
16	RX	T_{RDF}	RX data fall time		0		8
17	RX	T_{RDH}	RX data hold time (RCLK to RX data change)		5		
18	RX	T_{RDS} (See Note)	RX data setup time (RX data stable to the rising edge of RCLK)		See Note		
19	RENA	T_{DPL}	RENA low time		120		
20	RENA	T_{RENH}	RENA hold time after rising edge of RCLR		40		
21	CLSN	T_{CPH}	CLSN high time		80		
22	A/DAL	T_{DOFF}	Bus master driver disable after rising edge of $\overline{HOLD}$		0		50
23	A/DAL	T_{DON}	Bus master driver enable after falling edge of $\overline{HLDA}$		0		150
24	$\overline{HLDA}$	T_{HHA}	Delay to falling edge of $\overline{HLDA}$ from falling edge of $\overline{HOLD}$ (bus master)		0		
25	$\overline{RESET}$	T_{RW}	$\overline{RESET}$ pulse width low		200		
26	A/DAL	T_{CYCLE}	Read/write, address/data cycle time		600		
27	A	T_{XAS}	Address setup time to the falling edge of ALE		75		
28	A	T_{XAH}	Address hold time after the rising edge of $\overline{DAS}$		35		
29	DAL	T_{AS}	Address setup time to the falling edge of ALE		75		
30	DAL	T_{AH}	Address hold time after the falling edge of ALE		35		
31	DAL	T_{RDAS}	Data setup time to the rising edge of $\overline{DAS}$ (bus master read)		50		

NOTE: $T_{RDS\ (min)} = T_{RCT} - 25$ ns. Therefore, T_{RCT} = 100 ns when $T_{RDS\ (min)}$ = 75 ns.

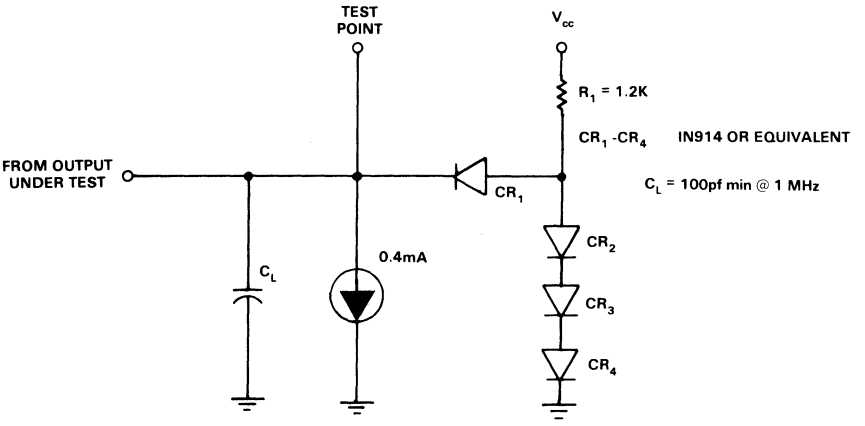
AC TIMING SPECIFICATIONS (Continued)

T_A = 0°C to 70°C, V_{CC} = +5 V ±5 percent, unless otherwise specified.

NO.	SIGNAL	SYMBOL	PARAMETER	TEST CONDITIONS	MIN (ns)	TYP (ns)	MAX (ns)
32	DAL	T_{RDAH}	Data hold time after the rising edge of $\overline{DAS}$ (bus master read)		0		
33	DAL	T_{DDAS}	Data setup time to the falling edge of $\overline{DAS}$ (bus master write)		0		
34	DAL	T_{WDS}	Data setup time to the rising edge of $\overline{DAS}$ (bus master write)		200		
35	DAL	T_{WDH}	Data hold time after the rising edge of $\overline{DAS}$ (bus master write)		35		
36	DAL	T_{SDO1}	Data driver delay after the falling edge of $\overline{DAS}$ (bus slave read)	(CSR 0,3, RAP)		400	
37	DAL	T_{SDO2}	Data driver delay after the falling edge of $\overline{DAS}$ (bus slave read)	(CSR 1,2)		1200	
38	DAL	T_{SRDH}	Data hold time after the rising edge of $\overline{DAS}$ (bus slave read)		0		35
39	DAL	T_{SWDH}	Data hold time after the rising edge of $\overline{DAS}$ (bus slave write)		0		
40	DAL	T_{SWDS}	Data setup time to the falling edge of $\overline{DAS}$ (bus slave write)		0		
41	ALE	T_{ALEW}	ALE width high		120		150
42	ALE	T_{DALE}	Delay from rising edge of $\overline{DAS}$ to the rising edge of ALE		70		
43	$\overline{DAS}$	T_{DSW}	$\overline{DAS}$ width low		200		
44	$\overline{DAS}$	T_{ADAS}	Delay from the falling edge of ALE to the falling edge of $\overline{DAS}$		80		130
45	$\overline{DAS}$	T_{RIDF}	Delay from the rising edge of $\overline{DALO}$ to the falling edge of $\overline{DAS}$ (bus master read)		15		
46	$\overline{DAS}$	T_{RDYS}	Delay from the falling edge of $\overline{READY}$ to the rising edge of $\overline{DAS}$	Taryd= 300 ns	75		250
47	$\overline{DALI}$	T_{ROIF}	Delay from the rising edge of $\overline{DALO}$ to the falling edge of $\overline{DALI}$ (bus master read)		15		
48	$\overline{DALI}$	T_{RIS}	$\overline{DALI}$ setup time to the rising edge of $\overline{DAS}$ (bus master read)		135		
49	$\overline{DALI}$	T_{RIH}	$\overline{DALI}$ hold time after the rising edge of $\overline{DAS}$ (bus master read)		0		
50	$\overline{DALI}$	T_{RIOF}	Delay from the rising edge of $\overline{DALI}$ to the falling edge of $\overline{DALO}$ (bus master read)		55		
51	$\overline{DALO}$	T_{OS}	$\overline{DALO}$ setup time to the falling edge of ALE (bus master read)		110		
52	$\overline{DALO}$	T_{ROH}	$\overline{DALO}$ hold time after the falling edge of ALE (bus master read)		35		
53	$\overline{DALO}$	T_{WDSI}	Delay from the rising edge of $\overline{DAS}$ to the rising edge of $\overline{DALO}$ (bus master write)		35		

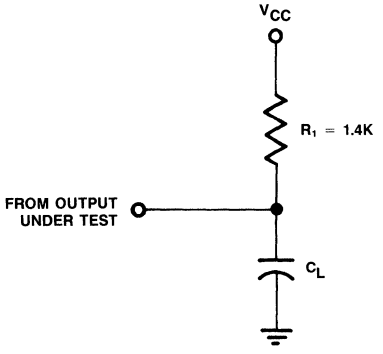
AC TIMING SPECIFICATIONS (Continued)
T_A = 0°C to 70°C, V_{CC} = +5 V ±5 percent, unless otherwise specified.

NO.	SIGNAL	SYMBOL	PARAMETER	TEST CONDITIONS	MIN (ns)	TYP (ns)	MAX (ns)
54	$\overline{CS}$	T_{CSH}	$\overline{CS}$ hold time after the rising edge of $\overline{DAS}$ (Bus slave)		0		
55	$\overline{CS}$	T_{CSS}	$\overline{CS}$ setup time to the falling edge of $\overline{DAS}$ (Bus slave)		0		
56	ADR	T_{SAH}	ADR hold time after the rising edge of $\overline{DAS}$ (Bus slave)		0		
57	ADR	T_{SAS}	ADR setup time to the falling edge of $\overline{DAS}$ (Bus slave)		0		
58	$\overline{READY}$	T_{ARYD}	Delay from the falling edge of ALE to the falling edge of $\overline{READY}$ to insure a minimum bus cycle time (600 ns)				80
59	$\overline{READY}$	T_{SRDS}	Data setup time to the falling edge of $\overline{READY}$ (Bus slave read)		75		
60	$\overline{READY}$	T_{RDYH}	$\overline{READY}$ hold time after the rising edge of $\overline{DAS}$ (Bus master)		0		
61	$\overline{READY}$	T_{SRO1}	$\overline{READY}$ driver turn on after the falling edge of $\overline{DAS}$ (Bus slave)	(CSR 0,3, RAP)		600	
62	$\overline{READY}$	T_{SRO2}	$\overline{READY}$ driver turn on after the falling edge of $\overline{DAS}$ (Bus slave)	(CSR 1,2)		1400	
63	$\overline{READY}$	T_{SRYH}	$\overline{READY}$ hold time after the rising edge of $\overline{DAS}$ (Bus slave)		0		35
64	READ	T_{SRH}	READ hold time after the rising edge of $\overline{DAS}$ (Bus slave)		0		
65	READ	T_{SRS}	READ setup time to the falling edge of $\overline{DAS}$ (Bus slave)		0		



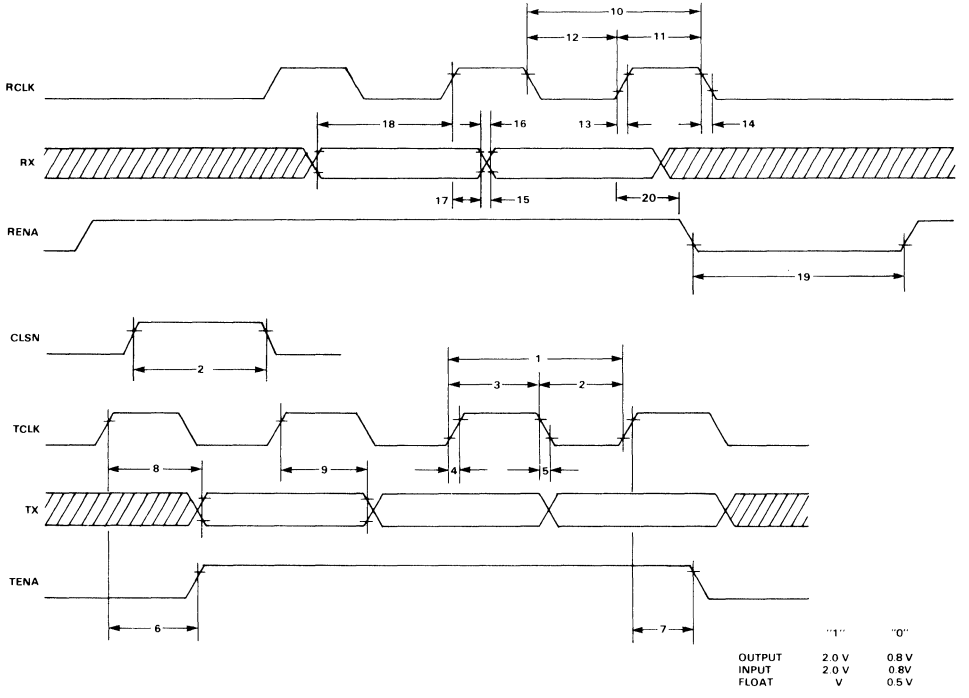
NOTE: This load is used on DAL00 through DAL15, READ, DALI, DALO, DAS, BM0, BM1, ALE/AS, A16 through A23, TENA, and TX.

Figure 5. Output Load Diagram



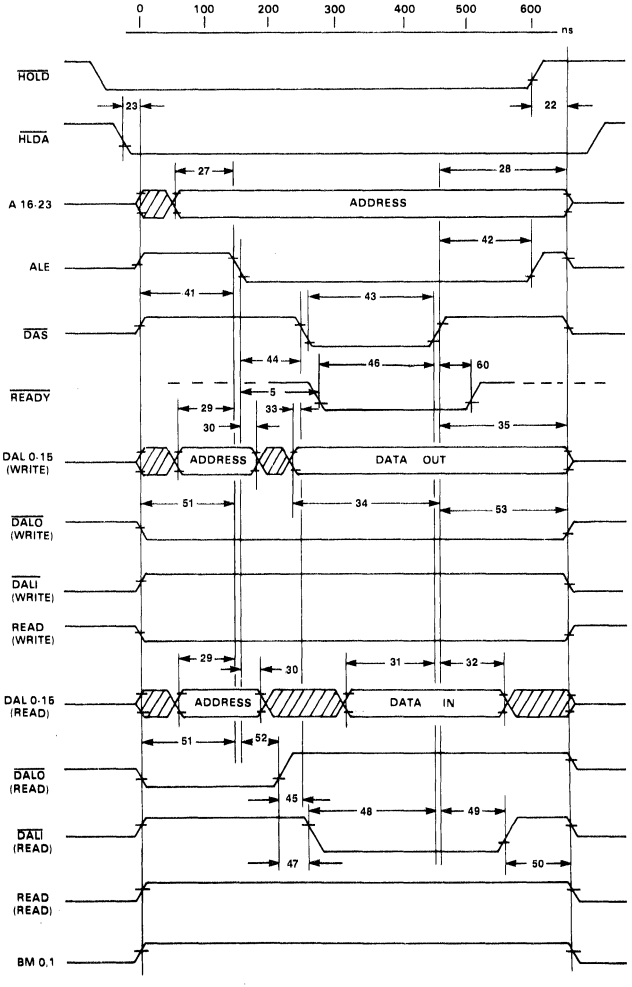
NOTE: This load is used on open drain outputs $\overline{\text{INTR}}$, HOLD/BUSRQ, and $\overline{\text{READY}}$.

Figure 6. Open Drain Output Load Diagram.



NOTE: Timing measurements are made at the following voltages unless otherwise specified.

Figure 7. Serial Link Timing Diagram - SIA Interface Signals



NOTE: The Bus Master cycle time will increase from a minimum of 600 ns in 100 ns steps until the slave device returns $\overline{\text{READY}}$.

Figure 8. LANCE Bus Master Timing Diagram

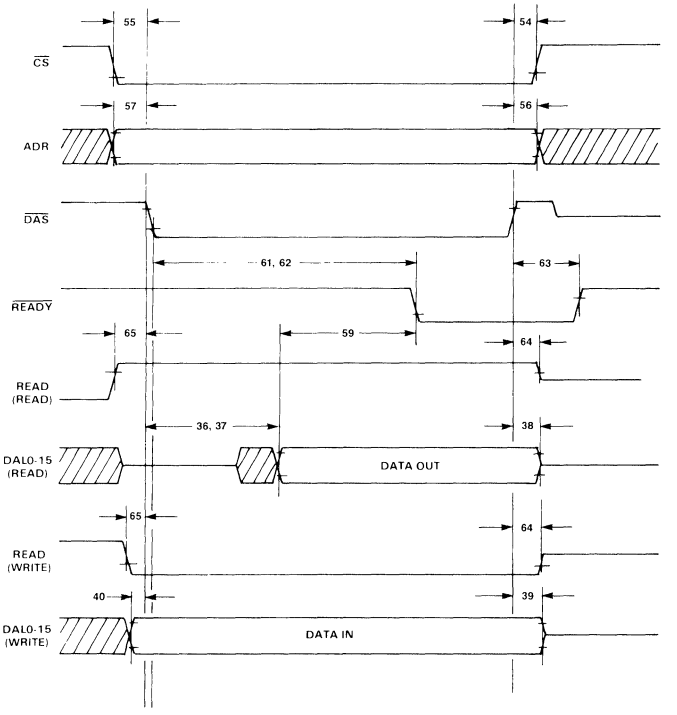


Figure 9. LANCE Bus Slave Timing Diagram

MK68591/2(P,J,N)

SERIAL INTERFACE ADAPTER (SIA)

COMMUNICATIONS PRODUCTS

FEATURES

- ☐ Compatible with Ethernet and IEEE-802.3 Specifications
- ☐ Crystal-controlled Manchester Encoder/Decoder
- ☐ Manchester Decoder acquires clock and data within six-bit times with an accuracy of ±3 ns.
- ☐ Guaranteed carrier and collision detection squelch threshold limits
  - — Carrier/collision detected for inputs more negative than −275 mV
  - — No carrier/collision for inputs more positive than −175 mV
- ☐ Input signal conditioning rejects transient noise
  - — Transients <10 ns for collision detector inputs
  - — Transients <20 ns for carrier detector inputs
- ☐ Receiver decodes Manchester data with up to ±20 ns clock jitter (at 10 MHz)
- ☐ TTL-compatible host interface
- ☐ Transmit oscillator accuracy ±0.01% (without adjustments)

GENERAL DESCRIPTION

The MK68591/2 Serial Interface Adapter (SIA) is a Manchester Encoder/Decoder compatible with Ethernet and IEEE-802.3 specifications. In an Ethernet/IEEE-802.3 application, the MK68591/2 interfaces the MK68590 Local Area Network Controller for Ethernet (LANCE™) to the Ethernet transceiver cable, acquires clock and data within 6 bit-times and decodes Manchester data up to ± 20 ns phase jitter at 10 MHz. SIA provides both guaranteed signal threshold limits and transient noise suppression circuitry in both data and collision paths to minimize false start conditions.

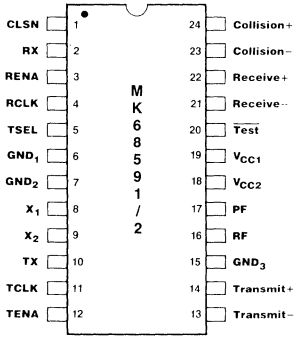


Figure 1. Pin Assignments

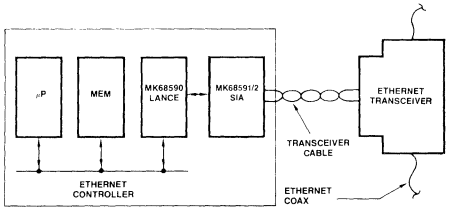


Figure 2. Typical Ethernet Node

LANCE is a trademark of Thomson Components - Mostek Corporation.

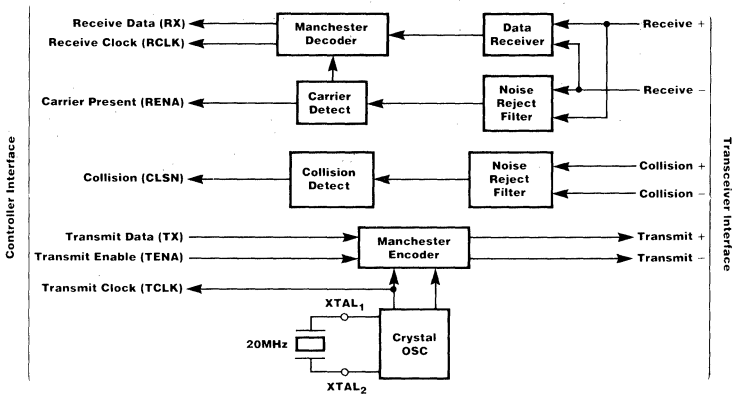


Figure 3. MK68591/2 Block Diagram

PIN DESCRIPTION

CLSN — **Collision (output).** A TTL active high output. Signals at the Collision ± terminals meeting threshold and pulse width requirements will produce a logic high at CLSN output. When no signal is present at Collision ±, CLSN output will be low.

RX — **Receive Data (output).** A MOS/TTL output, recovered data. When there is no signal at Receive ± and $\overline{\text{TEST}}$ is high, RX is high. RX is actuated with RCLK and remains activated until end of message. During reception, RX is synchronous with RCLK and changes after the rising edge of RCLK.

RENA — **Receive Enable (output).** A TTL active high output. When there is no signal at Receive ± and $\overline{\text{TEST}}$ is high, RENA is low. Signals at Receive± meeting threshold and pulse width requirements will produce a logic high at RENA. When Receive ± becomes idle, RENA returns to the low state synchronous with the rising edge of RCLK.

RCLK — **Receive Clock (output).** A MOS/TTL output recovered clock. When there is no signal at Receive ± and $\overline{\text{TEST}}$ is high, RCLK is low. RCLK is activated after the third negative data transition at Receive ±, and remains active until end of message. When $\overline{\text{TEST}}$ is low, RCLK is enabled.

TX — **Transmit (input).** TTL compatible input. When TENA is high, signals at TX meeting setup and hold time to TLCK will be encoded as normal Manchester at Transmit + and Transmit –.

TX High: Transmit + is negative with respect to Transmit – for first half of data bit cell.

TX Low: Transmit + is positive with respect to Transmit – for first half of data bit cell.

TENA — **Transmit Enable (input).** TTL compatible input. Active high data encoder enable. Signals meeting setup and hold time to TCLK allow encoding of Manchester data from TX to Transmit + and Transmit –.

TCLK — **Transmit Clock (output).** MOS/TTL output. TCLK provides symmetrical high and low clock signals at data rate for reference timing of data to be encoded. It also provides clock signals for the controller chip (MK68590 LANCE) and an internal timing reference for receive path voltage controlled oscillators.

Transmit +
Transmit – — **Transmit (outputs).** A differential line output. This line pair is intended to operate into terminated transmission lines. For signals meeting setup and hold time to TCLK at TENA and TX Manchester clock and data are outputted at Transmit +/Transmit –. When operating into a 78Ω terminated transmission line, signalling meetings the required output

levels and skew for both Ethernet and IEEE-802.3 drop cables.

Receive +
Receive −
Receiver (inputs). A differential input. A pair of internally biased line receivers consisting of a carrier detect receiver with offset threshold and noise filtering to detect the signal, and a data recovery receiver with no offset for Manchester data decoding.

Collision +
Collision −
Collision (inputs). A differential input. An internally biased line receiver input with offset threshold and noise filtering. Signals at Collision ± have no effect on data path functions.

TSEL
Transmit Mode Select. An open collector output and sense amplifier input.

TSEL Low: Idle transmit state Transmit + is positive with respect to Transmit −.

TSEL High: Idle transmit state Transmit + and Transmit − are equal, providing "zero" differential to operate transformer coupled loads.

When connected with an RC network, TSEL is held low during transmission. At the end of transmission, the open collector output is disabled, allowing TSEL to rise and provide a smooth transmission from logic high to "zero" differential idle. Delay and output return to zero are externally controlled by the RC time constant TSEL.

X_1, X_2
Biased Crystal Oscillator. X_1 is the input and X_2 is the bypass port. When connected for crystal operation, the system clock which appears at TCLK is half the frequency of the crystal oscillator. X_1 may be driven from an external source of two times the data rate.

RF
Frequency Setting Voltage Controlled Oscillator (V_{CO}) Loop Filter. This loop filter output is a reference voltage for the receive path phase detector. It also is a reference for timing noise immunity circuits in the collision and receive enable path. Nominal reference V_{CO} gain is 1.25 TCLK frequency MHz/V.

PF
Receive Path V_{CO} Phase Lock Loop Filter. This loop filter input is the control for receive path loop damping. Frequency of the receive V_{CO} is internally limited to transmit frequency ± 12%. Nominal receive V_{CO} gain is 0.25 reference V_{CO} gain MHz/V.

$\overline{TEST}$
Test Control (input). A static input that is connected to V_{CC} for normal MK68591/2 operation and to ground for testing of receive path function. When $\overline{TEST}$ is grounded, RCLK and RX are enabled so that receive path loop may be functionally tested.

GND_1 **High Current Ground**

GND_2 **Logic Ground**

GND_3 **Voltage Controlled Oscillator Ground**

V_{CC1} **High Current and Logic Supply**

V_{CC2} **Voltage Controlled Oscillator Supply**

FUNCTIONAL DESCRIPTION

The MK68591/2 Serial Interface Adapter (SIA) has three basic functions. It is a Manchester Encoder/line driver in the transmit path, a Manchester Decoder with noise filtering and quick lock-on characteristics in the receive path, and a signal detect/converter (10 MHz differential to TTL) in the collision path. In addition, the SIA provides the interface between the TTL logic environment of LANCE and the differential signaling environment in the transceiver cable.

TRANSMIT PATH

The transmit section encodes separate clock and NRZ* data input signals meeting the set up and hold time to TCLK at TENA and TX, into a standard Manchester II serial bit stream. The transmit outputs (Transmit +/ Transmit −) are designed to operate into terminated transmission lines. When operating into a 78Ω terminated transmission line, signaling meets the required output levels and skew for both Ethernet and IEEE-802.3.

Transmitter Timing and Operation

A 20 MHz fundamental mode crystal oscillator provides the basic timing reference in the SIA. It is divided by two to create the transmit clock reference (TCLK). Both 20 MHz and 10 MHz clocks are fed into the Manchester Encoder to generate the transitions in the encoded data stream. The 10 MHz clock, TCLK, is used by the SIA to internally synchronize transmit data (TX) and transmit

*Non-Return-to-Zero

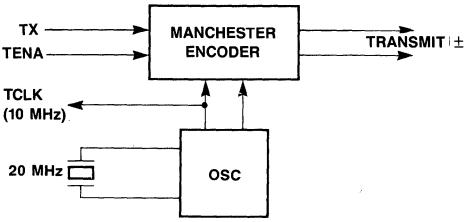


Figure 4. Transmit Section

enable (TENA). TCLK is also used as a stable bit-rate clock by the receive section of the SIA and by other devices in the system (the MK68590 LANCE uses TCLK to drive its internal state machine). The oscillator may use an external 0.005% crystal or an external TTL level input as a reference. Transmit accuracy of 0.01% is achieved (no external adjustments are required).

TENA is activated when the first bit of data is made available on TX. As long as TENA remains high, signals at TX will be encoded as Manchester and will appear at Transmit + and Transmit −. When TENA goes low, the differential transmit outputs go to one of the two idle states defined below:

- TSEL High: The idle state of Transmit +/ Transmit − yields "zero" differential to operate transformer coupled loads (see Figure 14a).

- TSEL Low: In this idle state, Transmit + is positive to Transmit − logical high (see Figure 14b).

RECEIVE PATH

The principle function of the receiver is the separation of the Manchester encoded data stream into clock and NRZ data.

Input Signal Conditioning

Before the data and clock can be separated, it must be determined whether there is "real" data or unwanted noise at the transceiver interface. The MK68591/2 SIA carrier detection receiver provides a static noise margin of −175 to −275 mV for received carrier detection. These DC thresholds assure that no signal more positive than −175 mV is ever decoded and that signals more negative than −275 mV are always decoded. Transient noise of less than 10ns duration in the collision path and 20 ns duration in the data path are also rejected.

This signal conditioning prevents unwanted idle noise on the transceiver cable from causing "false starts" in the receiver. This helps assure a valid response to "real" data.

The receiver section, shown in Figure 6, consists of two data paths. The receive data path is designed to be a zero threshold, high bandwidth receiver. The carrier detection receiver has an additional bias generator. Only data amplitudes larger than the bias level are interpreted as valid data. The noise rejection filter prevents noise transients of less than 20 ns from enabling the data receiver output. The collision detector similarly rejects noise transients of less than 10 ns.

Receiver Section Timing

Receive Enable (RENA) is the "carrier present" indication established when a signal of sufficient amplitude (V_{IDC}) and duration (t_{RPWR}) is present at the receive inputs. Receive Clock (RCLK) and Receive Data (RX) become available after the third negative data transition

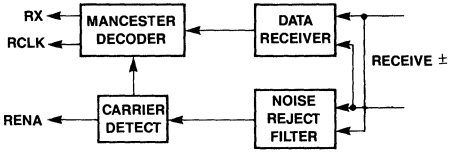


Figure 5. Receiver

at Receive +/ Receive − inputs, and stay active until the end of a packet. During reception, RX is synchronous with RCLK, changing after the rising edge of RCLK.

The receiver detects the end of a packet when the normal transition on the differential inputs cease. After the last low-to-high transition, RENA goes low and RCLK completes one last cycle, storing the last data bit. It then becomes and remains low (see Receive End of Packet Timing diagrams). When $\overline{\text{TEST}}$ is low, RCLK continues to run, tracking data (if available) or synchronize with TCLK.

Receive Clock Control

To insure quick capture of incoming data, the receiver phase-locked-loop is frequency locked to the transmit oscillator and it phase locks to incoming data edges. Clock and data are available within 6 bit times (accurate to within ±3 ns). The SIA will decode jittered data of up to ±20 ns (see Figure 7).

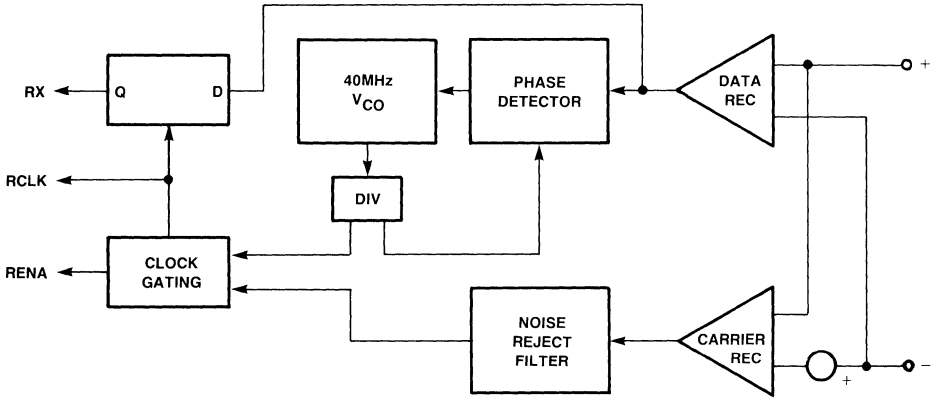


Figure 6. Receiver Section Detail

Differential I/O Terminations

The differential input for the Manchester data (receive ±) is externally terminated by two 40.2 Ω±1% resistors and one optional common mode bypass capacitor. The differential input impedance Z_{IDF} and the common mode input Z_{ICM} are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. The Collision ± differential input is terminated in exactly the same way as the receive input (see Figure 8).

Collision Detection

The Ethernet Transceiver detects collisions on the Ethernet and generates a 10MHz signal on the transceiver cable (Collision +/ Collision –). This collision signal passes through an input stage which assures signal levels and pulse duration. When the signal is detected by the SIA, the SIA sets the CLSN line high. This condition continues for approximately 190ns after the last low-to-high transition on Collision +/ Collision –.

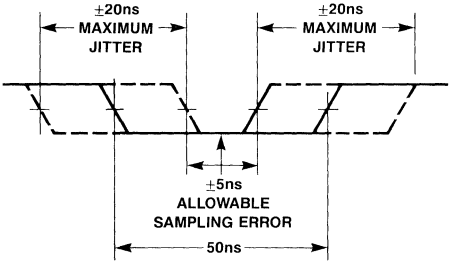
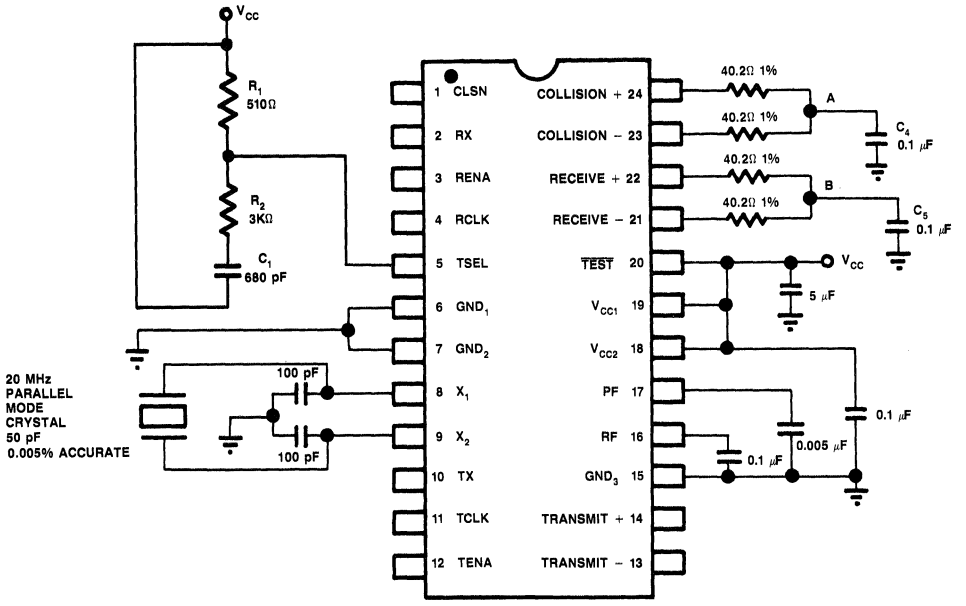


Figure 7. Maximum Jitter Impact On Sampling



NOTES:
1. Connect R_1, R_2, C_1 for 0 differential nontransmit. Connect to ground for logic 1 differential nontransmit.
2. Pin 20 shown for normal device operation.
3. Nodes A and B may be connected directly to ground for proper decoder operations, or to the common mode bypass C_4 and C_5. Some direct coupled transceivers require C_4 and C_5 to ground for proper operation.

Figure 8. MK68591/2 External Component Diagram

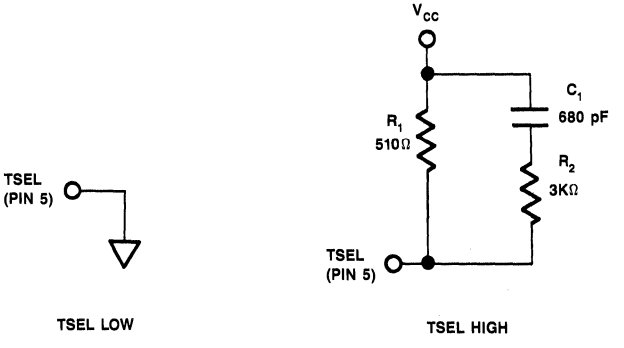


Figure 9. Transmit Mode Select (TSEL) Connection

ABSOLUTE MAXIMUM RATINGS*

Storage Temperature ..... −65 to +150°C
Temperature (Ambient) Under Bias ..... 0 to 70°C
Supply Voltage to Ground Potential Continuous ..... +7.0V
DC Voltage Applied to Outputs for High Output State ..... −0.5 to +V_{CC} max
DC Input Voltage (Logic Inputs) ..... +5.5V
DC Input Voltage (Receive/Collision) ..... −6 to +6 V
Transmit ± Output Current ..... −50 to +5 mA
DC Output Current, Into Outputs ..... 100 mA
DC Input Current (Logic Inputs) ..... ±30 mA

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

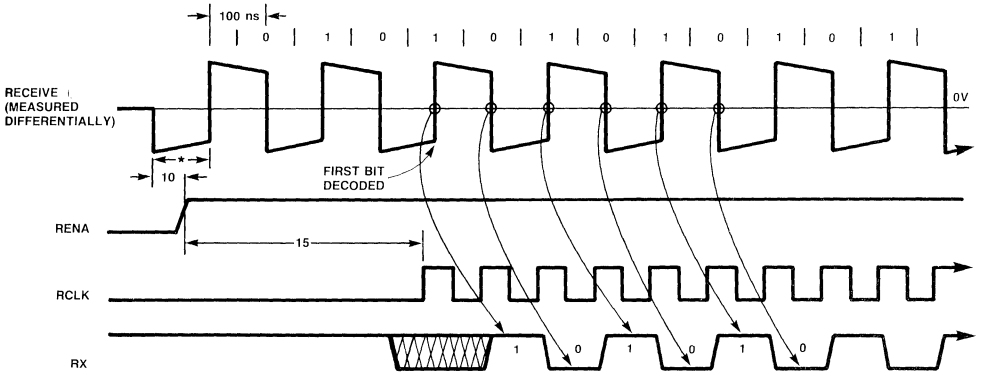
DC ELECTRICAL CHARACTERISTICS OVER OPERATING RANGE The following conditions apply unless otherwise specified:
T_A = 0 to 70°C, V_{CC} = 5.0 V±10 percent, MIN = 4.5 V, MAX = 5.5 V, period of crystal oscillator (T_{OSC}) = 50 ns

Parameters	Description		Test Conditions	Min	Typ	Max	Units
V_{OH}	Output High Voltage RX, RENA, CLSN, TCLK, RCLK		I_{OH} = −1.0 mA	2.4	3.4		V
V_{OL}	Output Low Voltage RCLK, TCLK, RENA, RX, CLSN, TSEL		I_{OL} = 16 mA,		0.36	0.5	V
			I_{OL} = 1 mA		0.25	0.4	
V_{OD}	Differential Output Voltage (Transmit +) − (Transmit −)	V_0	R_L = 78 Ω Figure 19	550	670	770	mV
		$\overline{V_0}$		−550	−670	−770	
$V_{OD\ OFF}$	Transmit Differential Output Idle Voltage		R_L =78 Ω Figure 19 TSEL = HIGH	−20	0.5	20	mV
$I_{OD\ OFF}$	Transmit Differential Output Idle Current			−0.5	±0.1	0.5	mA
V_{CMT}	Common Mode Output Transmit Voltage		Figure 19 R_L = 78 Ω	0	2.5	5	V
V_{ODI}	Differential Output Voltage Imbalance (Transmit ±) $\lvert\lvert V_0 \rvert - \lvert \overline{V_0} \rvert\rvert$				5	20	mV
V_{IH}	Input High Voltage TTL			2.0			V
I_{IH}	Input High Current TTL		V_{CC} = Max, V_{IN} = 2.7 V			+50	μA
V_{IL}	Input Low Voltage TTL					0.8	V
I_{IL}	Input Low Current TTL		V_{CC} = Max, V_{IN} = 0.4 V		−270	−400	μA
V_{IRD}	Differential Input Threshold (Rec Data)		Figure 20	−25	0	+25	mV
V_{IDC}	Differential Input Threshold (Carrier/Collision ±)		Figure 20	−175	−225	−275	mV
I_{CC}	Power Supply Current		t_{OSC} = 50 ns		125	180	mA
			t_{OSC} = 50 ns, T_A = Max			160	
V_{IB}	Input Breakdown Voltage V_I = +5.5 (TX, TENA, $\overline{TEST}$)		I_I = 1 mA	5.5			V
V_{IC}	Input Clamp Voltage		I_{IN} = −18 mA			−1.2	V
I_{SCO}	RX, TCLK, CLSN, RENA, RCLK Short Circuit Current			−40	−80	−150	mA
R_{IDF}	Differential Input Resistance		V_{CC} = 0 to Max	6k	8.4k	13k	ohm
R_{ICM}	Common Mode Input Resistance		V_{CC} = 0 to Max	1.5k	2.1k	7.5k	ohm
V_{ICM}	Receive and Collision Input Bias Voltage		I_{IN} = 0	1.5	3.5	4.2	V
I_{ILD}	Receive and Collision Input Low Curent		V_{IN} = −1 V	−0.6	−1.06	−1.64	mA
I_{IHD}	Receive and Collision Input High Current		V_{IN} = 6 V	+0.4	+0.6	+1.10	mA
I_{IHZ}	Receive and Collision Input High Current		V_{CC} = 0, V_{IN} = +6 V	0.4	1.28	1.86	mA

SWITCHING CHARACTERISTICS OVER OPERATING RANGE The following conditions apply unless otherwise specified: T_A = 0 to 70°C, V_{CC} = 5.0 V±10 percent, MIN = 4.5 V, MAX = 5.5 V, T_{OSC} = 50 ns

#	Signal	Parameters	Description	Test Conditions	Min	Typ	Max	Units
RECEIVER SPECIFICATION								
1	RCLK	t_{RCT}	RCLK Cycle Time	C_L = 50 pF Figure 17a (See note)	85	100	118	ns
2	RCLK	t_{RCH}	RCLK High Time		38	50		ns
3	RCLK	t_{RCL}	RCLK Low Time		38	50		ns
4	RCLK	t_{RCR}	RCLK Rise Time			2.5	8	ns
5	RCLK	t_{RCF}	RCLK Fall Time			2.5	8	ns
6	RX	t_{RDR}	RX Rise Time			2.5	8	ns
7	RX	t_{RDF}	RX Fall Time			2.5	8	ns
8	RX	t_{RDH}	RX Hold Time (RCLK to RX Change)		5	8		ns
9	RX	t_{RDS}	RX Prop Delay (RCLK to RX Stable)			8	25	ns
10	RENA	t_{DPH}	RENA Turn-On Delay (V_{IDC} Max on Receive ± to $RENA_H$)	Figures 10, 16a, and 20		50	80	ns
11	RENA	t_{DPO}	RENA Turn-Off Delay (V_{IDC} Min on Receive ± to $RENA_L$)	Figures 11 and 20		265	300	ns
12	RENA	t_{DPL}	RENA Low Time	Figure 11	120	200		ns
13	Rec ±	t_{RPWR}	Receive ± Input Pulse Width to Reject (Input<V_{IDC} Min)	Figures 16a and 20		30	20	ns
14	Rec ±	t_{RPWO}	Receive ± Input Pulse Width to Turn-On (Input>V_{IDC} Max)	Figures 16a and 20	45	30		ns
15	RCLK	t_{RLT}	Decoder Acquisition Time	Figure 10		390	450	ns
COLLISION SPECIFICATION								
16	Coll ±0	t_{CPWR}	Collision Input Pulse Width to Reject (Input<V_{IDC} Min)	Figures 16b and 20		18	10	ns
17	Coll ±	t_{CPWO}	Collision Input Pulse Width to Turn-On (Collision ± Exceeds V_{IDC} Max)		26	18		ns
18	Coll ±	t_{CPWE}	Collision Input to Turn-Off CLSN (Input<V_{IDC} Max)		80	117		ns
19	Coll ±	t_{CPWN}	Collision Input to Not Turn-Off CLSN (Input>V_{IDC} Min)			117	160	ns
20	CLSN	t_{CPH}	CLSN Turn-On Delay (V_{IDC} Max on Collision ± to $CLSN_H$)	Figures 15, 16b, and 20		33	50	ns
21	CLSN	t_{CPO}	CLSN Turn-Off Delay (V_{IDC} Min on Collision ± to $CLSN_L$)			133	160	ns
TRANSMITTER SPECIFICATION								
22	TCLK	t_{TCL}	TCLK Low Time	t_{OSC} = 50 ns Figures 17b and 18	45	50	55	ns
23	TCLK	t_{TCH}	TCLK High Time		45	50	55	ns
24	TCLK	t_{TCR}	TCLK Rise Time			2.5	8	ns
25	TCLK	t_{TCF}	TCLK Fall Time			2.5	8	ns
26	TX, TENA	t_{TDS}, t_{TES}	TX and TENA Setup Time	Figures 13, 14a, 14b, and 17b	5	1.1		ns
27	TX, TENA	t_{TDH}, t_{TEH}	TX and TENA Hold Time		5	−1.1		ns
28	TX ±	t_{TOCE}	Transmit ± Output, (Bit Cell Center to Edge)	Figures 14a, 14b, and 19	49.5	50	50.5	ns
29	TCLK	t_{OD}	TCLK High to Transmit ± Output			80	100	ns
30	TX ±	t_{TOR}	Transmit ± Output Rise Time	20 through 80 percent Figure 19		2	4	ns
31	TX ±	t_{TOF}	Transmit ± Output Fall Time			2	4	ns
32	TX ±	V_{OD}	Undershoot Voltage at Zero Differential Point on Transmit Return to Zero (End of Message)	Figure 14a			−100	mV

NOTE:
Assumes equal capacitance loading on RCLK and RX.



* PULSE WIDTH OF ≥ 45 ns IS ALWAYS RECOGNIZED.
HOWEVER, PULSE WIDTH OF ≤ 20 ns IS REJECTED.

Figure 10. Receiver Timing — Start of Packet

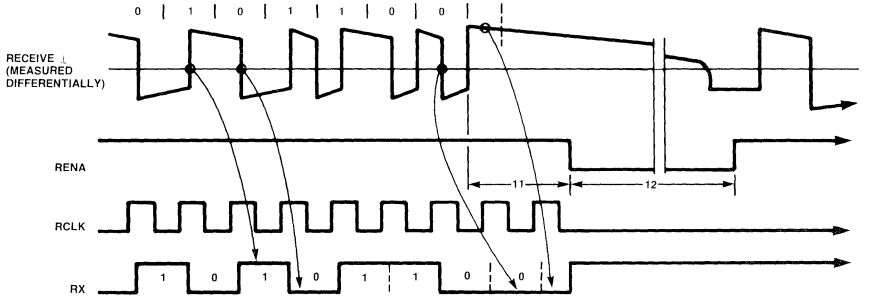


Figure 11. Receiver Timing — End of Packet (Last Bit = 0)

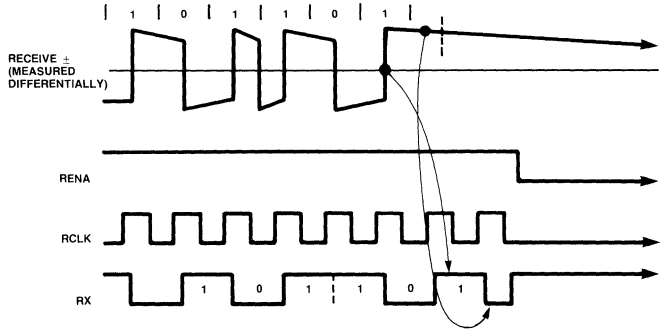


Figure 12. Receiver Timing — End of Packet (Last Bit = 1)

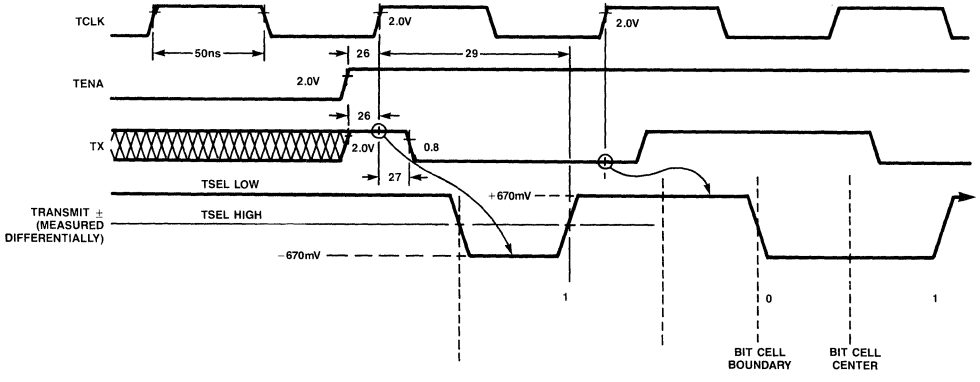


Figure 13. Transmitter Timing — Start of Transmission (TSEL Low, TSEL High)

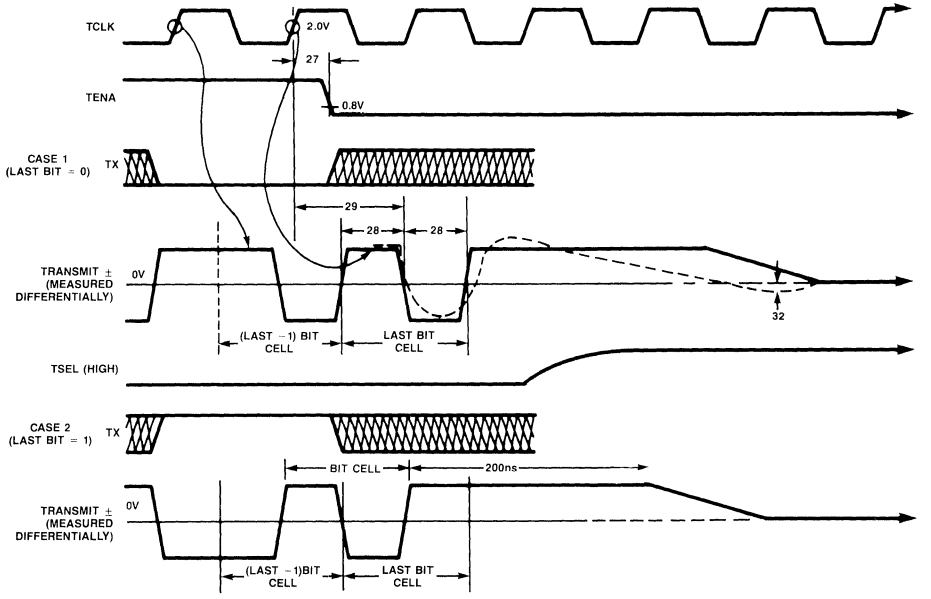


Figure 14a. Transmitter Timing — End of Transmission (TSEL High)

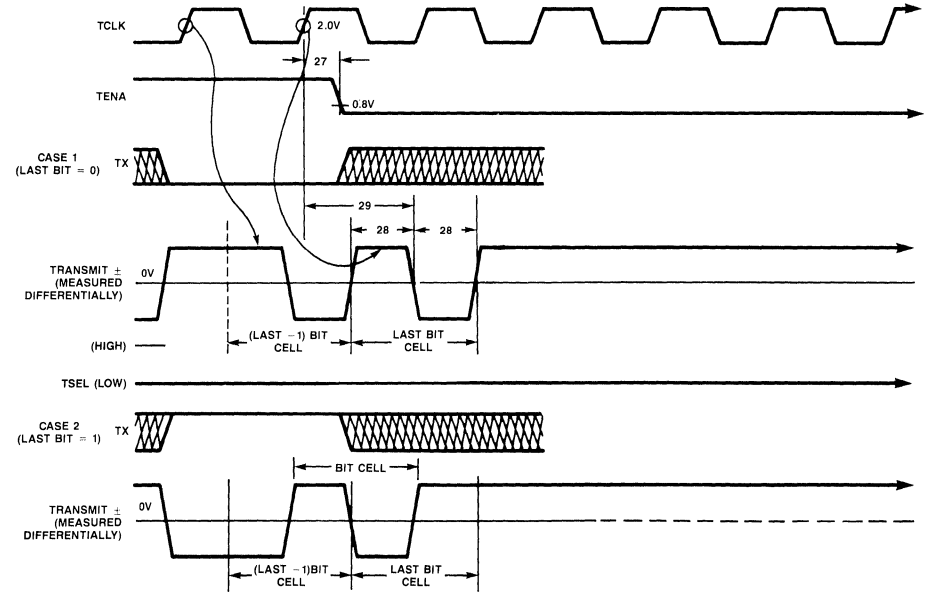
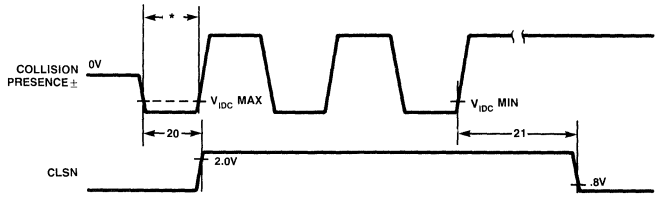


Figure 14b. Transmitter Timing — End of Transmission (TSEL Low)



* PULSE WIDTH OF −26 ns IS GUARANTEED TO BE RECOGNIZED: HOWEVER, PULSE WIDTH OF ≤ 10ns IS REJECTED

Figure 15. Collision Timing

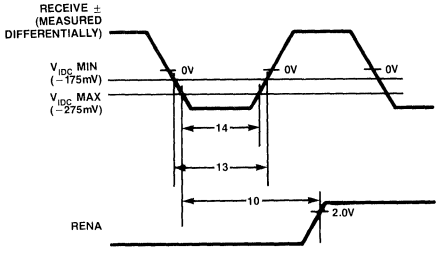


Figure 16a. Receive ± Input Pulse Width Timing

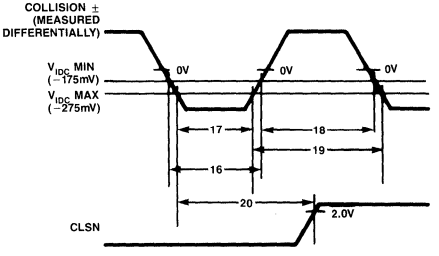


Figure 16b. Collision ± Input Pulse Width Timing

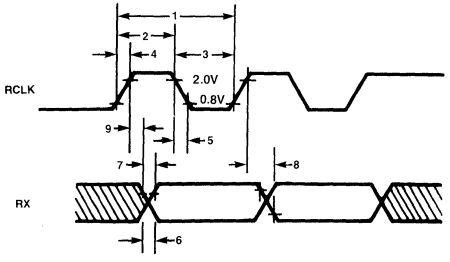


Figure 17a. RCLK and RX Timing

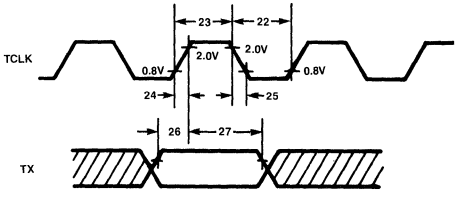


Figure 17b. TCLK and TX Timing

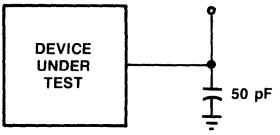


Figure 18. Test Load For RX, RENA, and TCLK

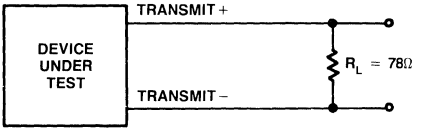


Figure 19. Transmit ± Output Test Circuit

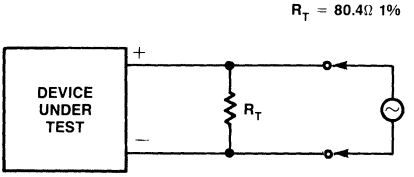
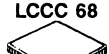


Figure 20. Receive ± and Collision ± Input Test Circuit

CHAPTER 9 - SURFACE MOUNTED DEVICES

LCCC SELECTION GUIDE

6800 MICROPROCESSORS	LCCC 28  CB-707	LCCC 44  CB-708	LCCC 52  CB-709	LCCC 68  CB-710	Page
EF6802		●			1-3
EF6809		●			1-111

6800 PERIPHERALS

EF6821		●			2-3
EF6840	●				2-15
EF6850	●				2-31
EF6854	●				2-41

68000 MICROPROCESSORS

TS68000				●	3-3
TS68008			●		3-95

68000 PERIPHERALS

MK68230			●		4-3
TS68HC901			●		4-59

CRT CONTROLLER

TS68483				●	6-97

68HC200/68200 - ROMLESS MCU

MK68HC200/68200			●		3-265
MK68HC200E/68200E				● (84)	

PLCC SELECTION GUIDE

6800 MICROPROCESSORS	PLCC 28 CB-520	PLCC 44  CB-521	PLCC 52  CB-522	PLCC 68  CB-523	Page
EF6802		●			1-3
EF6803		●			1-27
EF6803U4		●			1-67
EF6809		●			1-111
EF6809E		●			1-151

6800 PERIPHERALS

EF6821		●			2-3
EF6840	●				2-15
EF6850	●				2-31
EF6854	●				2-41

68000 MICROPROCESSORS

TS68000				●	3-3
TS68008			●		3-95

68000 PERIPHERALS

MK68230			●		4-3
MK68901			●		4-29
TS68HC901			●		4-59

CRT CONTROLLER

EF9345		●			6-3
EF9369	●				6-81
TS68483				●	6-97
TS68494		●			6-143

68HC200/68200 ROMLESS MCU

MK68HC200/68200			●		3-265
MK68HC200E/68200E				● (84)	

SURFACE MOUNTED DEVICES: today's solution for state-of-the-art system designs.

Today's trend toward light weight system designs with high component density allows Surface Mounting Technology to revolutionize manufacturing in the Electronics industry.

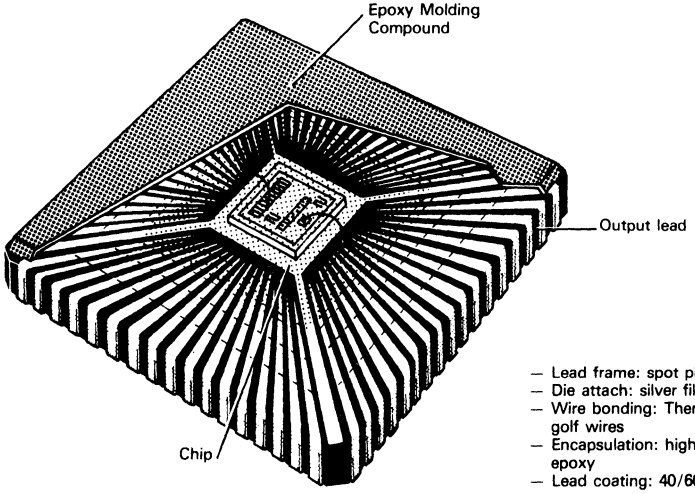
Reduction in board assembly cost by as much as 40% and in board size by as much as 50% is a goal that can be reached throught the utilization of Surface Mounted Devices:
Active Semiconductors (SO IC's) - SO Discretes and Chip carriers as well as passive resistors and capacitor chips.

Today system designers can select package outlines that meet state-of-the-art weight/space ratio requirements while enhancing electrical performance.

By 1990, through widely accepted data, one can foresee above 50% of the world wide demand to be Surface Mounted Devices.

Included in the THOMSON SEMICONDUCTEURS family of Surface Mounted Devices are integrated circuits (SO8 - SO28), Plastic Leaded Chip Carriers (PLCC) from 18 to 84 pins, Leadless Ceramic Chip Carrier's (LCCC) from 20 to 68 pads, as well as discrete diodes and transistors in SOT 23 (TO-236).

Capacitor chips are also available in THOMSON LCC. Compared to conventional types THOMSON SC's SMD have the following features:
- Compact design enabling high packing density and significant reduction in board size and weight, for instance in consumer electronics, telecommunications and automotives.
- Easy and low cost handling through automated high speed pick and place machinery.
- Mounting capability on both sides of all types of substrates (Ceramic or PC boards) using all current methods, such as wave soldering reflow and vapor phase technics.
- Same electrical characteristics (same dice) as conventional packages, with improved high frequency, high speed switching performances due to lower lead inductance and capacitance.
- Optimized way to package VLSI circuits by utilizing plastic chip carrier along with SO packages, leading to a major advantage over chip and wire assembly processes.



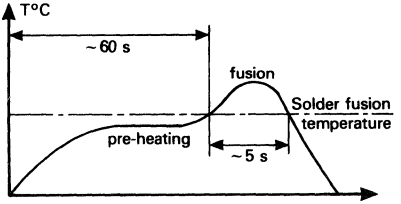
- Lead frame: spot plated copper
- Die attach: silver filled compound
- Wire bonding: Thermosonic ball bonding of golf wires
- Encapsulation: high performance non inflammable epoxy
- Lead coating: 40/60 Sn-Pb solder dipping.

MOUNTING METHODS

PLCC and LCCC may be mounted on either PC boards or substrates, employing different soldering methods.

Reflow process

• This method permits the soldering of all components in a single operation. In this case, solder cream is used either by screen printing or employing a pneumatic gun. After the application of solder cream the components are mounted and the circuit is guided through an infrared or convection oven which allows the solder to melt.
In order to avoid solder spreading risks and to position the components, the board should be pre-screen printed with an insulating material.



• An other way to solder coat the board is to dip it into a soldering bath or through wave soldering. Then, flux is applied by a brush or immersion and the components are placed.

The soldering is done by reflow through an oven as above.
The result obtained by both methods is identical. However, in this case, a complete tinning of the circuit is necessary.

Wave soldering process

The components are glued to substrate by means of an insulated glue. The board is dipped in flux and then goes through the wave soldering process.

Hot air soldering

In this system, the soldering iron must be replaced by a hot air nozzle. Employing this method, an improvement in solder temperature control is achieved.
It is also easy to correct some soldering failures with this method.

Heated collet

This method is used for rework and soldering PCC on the versus side of the board after wave solder.

Vapor phase soldering (VPS)

This method uses the heat of the vapor of a boiling inert fluorinated fluid. Soldering is accomplished by either screening tin layers or by electroplating solder paste.
Components are then mounted in appropriate locations and soldered to the circuit by dipping it into the vapor of boiling inert fluorinated fluid.
The main advantages of this method are:
• Uniform workpiece temperature whatever the difference in size of the components involved
• Accurate temperature without any elaborate control (temperature of boiling inert fluid)
• Dipping time may be very short and as a consequence better soldering results are obtained
• Soldering takes place in an oxygen-free atmosphere which permits:
— to use less active soldering flux, and
— to avoid soldering flux oxidation (easy cleaning of remaining flux).

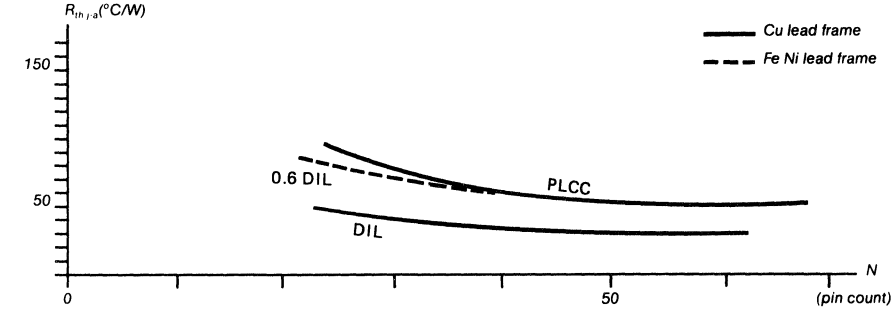
Remarks

Various types of high speed automatic mounting equipments for passive and active devices are available contributing to considerable reduction in production costs.

THERMAL CHARACTERISTICS

Thermal performances of SMD are dependent upon substrate material, size, mounting process, chip area, die attach and lead frame material characteristics.

For PLCC, copper is selected in order to improve the thermal characteristics. Therefore thermal resistance approaching that of a conventional Fe Ni 42 DIL can be obtained.



Junction-ambient air thermal resistance, components reported on classical PC board (FR 4, 1.3 mm thickness, 50 x 50 mm size).

Allowable power dissipation can reach 350 mW according to device type, refer to data sheet.

Trade Marks Registered ®

Thomson Components - Mostek Corporation reserves the right to make changes in specifications and other information at any time without prior notice. Information contained herein is believed to be correct, but is provided solely for guidance in product application and not as a warranty of any kind. Thomson Components - Mostek assumes no responsibility for use of this information, nor for any infringements of patents or other rights of third parties resulting from use of this information, nor for the use of any circuitry other than circuitry embodied in a Thomson Components - Mostek product. No license is granted under any patents or patent rights of Thomson Components - Mostek.

Thomson Components - Mostek Corporation reserves the right to make changes in specifications and other information at any time without prior notice. Information contained herein is believed to be correct, but is provided solely for guidance in product application and not as a warranty of any kind. Thomson Components - Mostek assumes no responsibility for use of this information, nor for any infringements of patents or other rights of third parties resulting from use of this information, nor for the use of any circuitry other than circuitry embodied in a Thomson Components - Mostek product. No license is granted under any patents or patent rights of Thomson Components - Mostek.

The "PRELIMINARY" designation on a Thomson - Mostek data sheet indicates that the product is not characterized. The specifications are subject to change, are based on design goals or preliminary part evaluation, and are not guaranteed. Thomson - Mostek or an authorized sales representative should be consulted for current information before using this product.

The "ADVANCE INFORMATION" designation on a Thomson - Mostek publication indicates that the item described is a prospective product, is not yet available, and that the specification goals have not yet been fully established. Production is anticipated but not guaranteed. The ADVANCE INFORMATION is an initial disclosure of a new product's features and description. The specifications are subject to change at any time without notice, are based on design goals, and are not guaranteed or warranted in any way. Thomson - Mostek or an authorized sales representative should be consulted for current information before using this product or basing any designs on this ADVANCE INFORMATION. No responsibility is assumed by Thomson - Mostek for its use, nor for any infringements of patents or trademarks or other rights of third parties resulting from its use. No license is granted under any patents, patent rights, or trademarks of Thomson - Mostek.

The "APPLICATION BRIEF" or "APPLICATION NOTE" designation on a Thomson-Mostek literature item indicates that the literature item contains information regarding Thomson-Mostek features and/or their varied applications. The information given in the APPLICATION BRIEF or APPLICATION NOTE is believed to be accurate and reliable; however, the information is subject to change and is not guaranteed. No responsibility is assumed by Thomson-Mostek for its use; nor for any infringements of patents and trademarks or other rights of third parties resulting from its use. No license is granted under any patents, patent rights, or trademarks of Thomson-Mostek.

1310 ELECTRONICS DRIVE / CARROLLTON. TEXAS 75006
PHONE: (214) 466-6000 / TELEX: 730643

PRINTED IN USA April 1987
Publication No. 4430244