Burroughs Corporation INTER-OFFICE CORRESPONDENCE

CORPORATE UNIT	LOCATION	DEPT.
Computer Systems Group	Santa Barbara Plant	Systems Eng. Section 6311

TO:

NAME	DATE
C. E. Wymore, Manager	30 March 1976

FROM	DEPT. & LOCATION
R. Matsuda	Systems Engineering Section 6311 - SBP

SUBJECT: C.C.

PRELIMINARY RELEASE OF THE M-PROCESSOR-3 ENGINEERING DESIGN SPECIFICATION

See Distribution attached

This preliminary release of the M-processor-3 (MP-3) Engineering Design Specification (EDS) is an editorial update of the 16 January version (Rev B). Also included in this release are the contents of the two inter-office memorandums:

1. Changes to M-Processor-3 EDS 3 February 1976, C.E. Wymore
2. Change to M-Processor-3 EDS: II 4 February 1976, C.E. Wymore

There were still other technical details which were unresolved as of the January Transfer of Information and those which have been resolved as of 26 March are also included in this release. Some of those issues are:

a. TIME-half microsecond time counter up to 24 bits (approximately 8 seconds);

b. PERM and PERP register relationship with respect to Halt, CD(3), Console display, and clearing;

c. MSSW - can be dynamically changed;

d. Operation of A and M - register during TAPE mode;

e. Skip on FA and BR comparison;

f. Diagnostic Read/Write Memory micro (11D) - echo variants

g. CA-RC spreader - 8 clocks for RC-RC, 4 clocks for RC-CA;

h. Error log message - finalization of the format;

i. Increment A - register - new micro; and

j. 18 Position Console Rotary Switch - redefinition.

The EDS at this stage still needs more work in the description of the three-phase or pipeline mechanism. Also, the general control logic philosophy and design needs more editing. Those figures for both sections must be drawn. Otherwise, for final release, the processor description will not be expanded. In many areas, the machine is much like the B1726 and it is assumed that the knowledge of it with this simple EDS is sufficient to describe the MP-3.

R Matsuda

R. Matsuda
Systems Eng. Section 6311

Attachments

M - P R O C E S S O R - 3

E N G I N E E R I N G D E S I G N S P E C I F I C A T I O N

2 2 1 5 9 8 9 1

R E V. B

AS OF 10:00 FRIDAY 16 JANUARY 76

REV. C

MARCH 26, 1976

TABLE OF CONTENTS

1.0 INTRODUCTION . . . . . PAGE 1
1.1 PURPOSE. . . . . PAGE 1
1.2 EDS PHILOSOPHY . . . . . PAGE 1
1.3 PRODUCT IDENTIFICATION . . . . . PAGE 1
1.4 RELATED SPECIFICATIONS, DOCUMENTS, AND DRAWINGS. . . . . PAGE 1
2.0 GENERAL DESCRIPTION. . . . . PAGE 3
2.1 THREE PHASE MICRO EXECUTION. . . . . PAGE 16
3.0 PRODUCT DESCRIPTION. . . . . PAGE 19
3.1 PROCESSOR REGISTERS. . . . . PAGE 19
3.1.1 M. . . . . PAGE 19
3.1.2 A-REGISTER. . . . . PAGE 19
3.1.3 TAS. . . . . PAGE 20
3.1.4 A-STACK. . . . . PAGE 20
3.1.5 X AND Y. . . . . PAGE 26
3.1.6 L. . . . . PAGE 26
3.1.7 T. . . . . PAGE 27
3.1.8 FA . . . . . PAGE 27
3.1.9 FB . . . . . PAGE 28
3.1.10 SCRATCHPAD. . . . . PAGE 28
3.1.11 FLCN. . . . . PAGE 29
3.1.12 BR AND LR . . . . . PAGE 29
3.1.13 C . . . . . PAGE 30
3.1.13.1 CP (CYF, CPU, CPL). . . . . PAGE 30
3.1.13.2 CA, CB, CC, CD. . . . . PAGE 30
3.1.14 MAXS. . . . . PAGE 32
3.1.15 U . . . . . PAGE 32
3.1.16 DATA. . . . . PAGE 33
3.1.17 CMND. . . . . PAGE 34
3.1.18 NULL. . . . . PAGE 34
3.1.19 CSW . . . . . PAGE 34
3.1.20 MSSW. . . . . PAGE 34
3.1.21 NULLA, NULLB, NULLC, NULLD, and NULLE . . . . . PAGE 35
3.1.22 RESERVED . . . . . PAGE 35
3.2 24-BIT FUNCTION BOX. . . . . PAGE 36
3.2.1 SUM. . . . . PAGE 36
3.2.2 DIFF . . . . . PAGE 37
3.2.3 XANY, XORY, XEOY . . . . . PAGE 37
3.2.4 CMPX, CMPY . . . . . PAGE 38
3.2.5 MSKX,MSKY. . . . . PAGE 38
3.2.6 BICN . . . . . PAGE 38
3.2.7 XYCN . . . . . PAGE 39
3.2.8 XYST . . . . . PAGE 39
3.2.9 INCN . . . . . PAGE 40
3.2.10 PERM (PARITY ERROR MEMORY). . . . . PAGE 40
3.2.11 PERP (Parity Error Processor) . . . . . PAGE 43
3.2.12 TIME. . . . . PAGE 47
3.3 4-BIT FUNCTION BOX . . . . . PAGE 47
3.4 M-INSTRUCTIONS . . . . . PAGE 47
3.4.1 REGISTER MOVE. . . . . PAGE 48
3.4.2 SCRATCHPAD MOVE. . . . . PAGE 48
3.4.3 SWAP F WITH DOUBLEPAD WORD . . . . . PAGE 49
3.4.4 STORE F INTO DOUBLEPAD WORD. . . . . PAGE 50
3.4.5 LOAD F FROM DOUBLEPAD WORD . . . . . PAGE 50
3.4.6 MOVE 8-BIT LITERAL . . . . . PAGE 50
3.4.7 MOVE 24-BIT LITERAL. . . . . PAGE 51
3.4.8 SWAP MEMORY. . . . . PAGE 51
3.4.9 READ/WRITE MEMORY. . . . . PAGE 52
3.4.10 COUNT FA/FL . . . . . PAGE 53
3.4.11 SCRATCHPAD RELATE FA. . . . . PAGE 54
3.4.12 EXTRACT FROM REGISTER T . . . . . PAGE 54
3.4.13 SHIFT/ROTATE REGISTER T LEFT. . . . . PAGE 54
3.4.14 SHIFT/ROTATE REGISTER X/Y LEFT/RIGHT. . . . . PAGE 55

3.4.15 SHIFT/ROTATE REGISTERS XY LEFT/RIGHT. . . . . PAGE 56
3.4.16 NORMALIZE X . . . . . PAGE 56
3.4.17 READ/WRITE CACHE. . . . . PAGE 56
3.4.18 CALL. . . . . PAGE 61
3.4.19 BRANCH. . . . . PAGE 62
3.4.20 BIAS. . . . . PAGE 62
3.4.21 SET CYF . . . . . PAGE 63
3.4.22 4-BIT MANIPULATE. . . . . PAGE 63
3.4.23 BIT TEST BRANCH FALSE . . . . . PAGE 64
3.4.24 BIT TEST BRANCH TRUE. . . . . PAGE 65
3.4.25 SKIP WHEN . . . . . PAGE 65
3.4.26 CLEAR REGISTERS . . . . . PAGE 67
3.4.27 BIND. . . . . PAGE 67
3.4.28 OVERLAY M-MEMORY. . . . . PAGE 67
3.4.29 DISPATCH. . . . . PAGE 68
3.4.30 CASSETTE CONTROL. . . . . PAGE 71
3.4.31 HALT. . . . . PAGE 71
3.4.32 NO OPERATION. . . . . PAGE 72
3.4.33 MONITOR . . . . . PAGE 72
3.4.34 NANO MOVE . . . . . PAGE 72
3.4.35 DIAGNOSTIC READ/WRITE MEMORY. . . . . PAGE 76
3.4.36 CLEAR CACHE . . . . . PAGE 83
3.4.37 INCREMENT A-REGISTER. . . . . PAGE 83
3.5 CONTROL PANEL OPERATIONS . . . . . PAGE 83
3.5.1 DIAGNOSTIC AND MAINTENANCE PANEL . . . . . PAGE 83
3.5.1.1 REGISTER SELECT SWITCH . . . . . PAGE 84
3.5.1.2 REGISTER GROUP SWITCH. . . . . PAGE 85
3.5.1.3 M-INPUT SWITCH . . . . . PAGE 85
3.5.1.4 INTERRUPT SWITCH . . . . . PAGE 86
3.5.1.5 LOAD SWITCH. . . . . PAGE 86
3.5.1.6 HALT SWITCH. . . . . PAGE 86
3.5.1.7 CLEAR SWITCH . . . . . PAGE 87
3.5.1.8 START SWITCH . . . . . PAGE 87
3.5.1.9 CASSETTE SELECT. . . . . PAGE 87
3.5.1.10 INCREMENT SWITCH. . . . . PAGE 87
3.5.1.11 SINGLE MICRO/CONTINUOUS . . . . . PAGE 87
3.5.1.12 DATA SWITCHES . . . . . PAGE 87
3.5.1.13 STATE LIGHT . . . . . PAGE 88
3.5.1.14 RUN LIGHT . . . . . PAGE 88
3.5.1.15 ERROR LIGHT . . . . . PAGE 88
3.5.1.16 OVER TEMPERATURE LIGHT. . . . . PAGE 88
3.5.1.17 DATA LIGHTS . . . . . PAGE 88
3.5.2 REMOTE OPERATIONAL PANEL . . . . . PAGE 88
3.5.2.1 BOT LIGHT. . . . . PAGE 89
3.5.2.2 START SWITCH AND LIGHT . . . . . PAGE 89
3.5.2.3 INTERRUPT SWITCH AND LIGHT . . . . . PAGE 89
3.5.2.4 CLEAR SWITCH . . . . . PAGE 89
3.5.2.5 TAPE/NORMAL SWITCH . . . . . PAGE 89
3.5.3 LOAD/DISPLAY OPERATIONS. . . . . PAGE 89
3.6 CACHE. . . . . PAGE 94
3.7 PROCESSOR CONTROL LOGIC. . . . . PAGE 94
3.8.1 INSTRUCTION EXECUTION LOGIC (IEC). . . . . PAGE 95
3.8.2 PROCESSOR OPERATIONAL CONTROLS . . . . . PAGE 95
3.8.2 Cont'd . . . . . PAGE 97
3.9 PROCESSOR OPERATION CONTROL. . . . . PAGE 97
3.9.1 CONSOLE REG SELECT ENCODERS. . . . . PAGE 97
3.9.2 CONSOLE SWITCH STATES ENCODER. . . . . PAGE 98
3.9.3 OPERATING MODE INTERLOCK . . . . . PAGE 100
3.9.4 PROCESSOR OPERATION CONTROL ROM (POCR) . . . . . PAGE 100
3.9.5 OPERATING PHASE COUNTER AND QUIESCENT PHASE DETECTOR . PAGE 101
3.9.6 POC DECODE INHIBIT . . . . . PAGE 101
3.9.0 INSTRUCTION EXECUTION CONTROL. . . . . PAGE 101
3.10.1.0 MICRO TO NANO DECODE. . . . . PAGE 104
3.10.1.1.0 NANO PROGRAM CONTROL STORE (NPCS) . . . . . PAGE 104

3.10.1.1.1 NANO PROGRAM WORD . . . . . PAGE 104
3.10.1.1.2 NANO EXECUTION CONTROL. . . . . PAGE 105
3.10.1.1.3 GENERATING ADDRESSES TO NANO PROGRAM CONTROL STORE PAGE 105
3.10.1.1.4 SEQUENCE NUMBER CONTROLS. . . . . PAGE 106
3.10.2.0 NANO EXECUTION CONTROL. . . . . PAGE 106
3.10.2.1 NANO STATE CONTROLLER . . . . . PAGE 106
3.11 THREE PHASE DURING NORMAL RUN, TAPE AND STEP MODES. . . PAGE 107
4.0 M-INSTRUCTION TIMING . . . . . PAGE 112
4.1 M-Instruction Performance. . . . . PAGE 115

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

1.0 INTRODUCTION

This EDS defines the functional characteristics of the B1820 M-Processor-3, hereinafter called "MP-3."

1.1 PURPOSE

This EDS is prepared by Systems Engineering as a vehicle for inter-designer communication and project control. Its intended audience is Systems Engineering, Diagnostics, and Software. It serves as input to the Product Spec which is exposed to other activities.

1.2 EDS PHILOSOPHY

The present state of this document represents the hardware design as conceived implementing the inferred product specification. It will be periodically updated and once the design has been completed, signified by the engineering release to manufacturing, this cocument will reflect that machine and will not change.

1.3 PRODUCT IDENTIFICATION

2212 8631	M-PROCESSOR-3
2212 8623	CABINET-5 (control panel is part of cabinet)

1.4 RELATED SPECIFICATIONS, DOCUMENTS, AND DRAWINGS

P.S. #	NAME
1904 5681	B1700/1800 SYSTEM INDEX
1913 1739	B1700/1800 CENTRAL SYSTEM
2212 9001	MP-3
2212 9019	CP-4

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
M-PROCESSOR-3
E.D.S. #2215 9891

S.D.S. #	NAME
2216 0683	I/O Base-3
2216 0691	B1820 System

S.P.S. #	NAME
2216 1962	MP-2
2216 2762	MBU-3

E.D.S. #	NAME
2215 7408	1x16Kx22 RAM Storage Board
2215 8513	B1820 Clock System

DOCUMENT	NAME
tbs	

DRAWING	NAME
tbs	

NOTE: A-SIZE REDUCTIONS OF "*"-ED DRAWINGS
---- ARE INCLUDED IN THIS EDS.

2.0 GENERAL DESCRIPTION

MP-3 provides the combinatorial and arithmetic portion of the system along with other registers and hard storage that are appropriate for efficient operation.

MP-3 provides hardware sensitivity to a set of low-level micro-functions which are used in a program string to fetch and execute instructions. This micro-program is contained in a local high-speed cache memory, backed up by a somewhat slower but larger main memory (B1800 S-memory) or in both. Cache is an integral 4K byte memory.

Included in MP-3 are registers and pseudo registers which are addressable by the individual micro-operators.

The registers are normally addressed by a 4-bit group (row) number and a 2-bit select (column) number as shown in Table 1.

Some of the registers listed, such as the Pseudo Sum Register, can serve only as source registers while others are capable of serving both as source and destination registers. Also, some of the registers listed are actually subregisters which, although parts of larger registers, can be individually addressed and manipulated.

Table 2 summarizes the various conditions available by addressing particular pseudo source registers and actual registers; Figures 1 and 2 list the micro-instructions and their variants; and Figures 3-9 are diagrams of the major portions of MP-3.

GROUP NUMBER	SELECT NUMBER 0	1	2	3
0	TA	FU	X	SUM
1	TB	FT	Y	CMPX
2	TC	FLC	T	CMPY
3	TD	FLD	L	XANY
4	TE	FLE	A	XEOY
5	TF	FLF	M	MSKX
6	CA	BICN	BR	MSKY
7	CB	FLCN	LR	XORY
8	LA	NULLA	FA	DIFF
9	LB	RESERVED	FB	MAXS
10	LC	PERM	FL	NULLC
11	LD	PERP	TAS	U
12	LE	XYCN	CP	NULLD
13	LF	XYST	NULLB	DATA
14	CC	INCN	CSW	CMND
15	CD	MSSW	TIME	NULL

TABLE 1 M-PROCESSOR REGISTER SELECTION

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
M-PROCESSOR-3
E.D.S. #2215 9891

BICN:

3	2	1	0: LSB
LSUY	CYF	CYD	CYL

XYCN:

3	2	1	0:LSB
MSBX	X=Y	X<Y	X>Y

XYST:

3	2	1	0:LSB
LSUX	INT	Y neq 0	X neq 0

FLCN:

3	2	1	0:LSB
FL=SFL	FL>SFL	FL<SFL	FL neq 0

INCN:

3	2	1	0:LSB
PORT DEVICE MISSING	PORT HI PRIORITY	PORT INTERRUPT	PORT LOCKOUT

CC:

3	2	1	0:LSB
CONTROL PANEL STATE LAMP FLIP-FLOP	100 MSEC REAL TIME CLOCK INTERRUPT	I/O BUS SERVICE REQUEST INTERRUPT	CONTROL PNL INTERRUPT

CD:

3	2	1	0:LSB
MEMORY READ DATA ERROR INTERRUPT	MEMORY WRITE/SWAP ADDR (LR/BR CHECK) OUT OF BOUNDS OVERRIDE	MEMORY READ ADDR (LR/BR CHECK) OUT OF BNDS INTERRUPT	MEMORY WRITE/SWAP ADDR (LR/BR CHECK) OUT OF BOUNDS INTERRUPT

TABLE 2 SUMMARY OF REGISTER CONDITIONS

MSSW:

3	2	1	0:LSB
NULL BIT	NULL BIT	S1: MICRO SOURCE FROM CACHE OR FROZEN IN M-REGISTER	S0: MICRO SOURCE FROM S-MEMORY OR FROZEN IN M-REGISTER

PERM:

3	2	1	0:LSB
S-MEMORY MICRO-INSTRUCTION TIME-OUT	S-MEMORY FIELD OUT-OF-BOUNDS IN THE ADMINISTRATIVE MEMORY	S-MEMORY ERROR LOG HAS CHANGED	S-MEMORY UNCORRECTABLE ERROR DURING A PROCESSOR OPERATION

PERP:

3	2	1	0:LSB
CACHE DOUBLE HIT	CACHE KEY PARITY ERROR ON KEY A OR KEY B	PARITY ERROR ON THE WORD FETCHED TO THE M-REGISTER	CASSETTE READ ERROR WHICH CANNOT BE CORRECTED

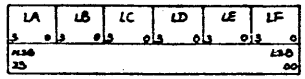
TABLE 2 (cont) SUMMARY OF REGISTER CONDITIONS

	MICRO NAME	MC 15	14	13	12	MD 11	10	9	8	ME 7	6	5	4	MF 3	2	1	0	VARIANTS: 0 00	1 01	2 10	3 11	4 100	5 101	6 110	7 111
1C	REGISTER MOVE	0	0	0	1	REGISTER 1 GROUP SOURCE REGISTER				REG. 1 SELECT		REG. 2 SELECT		REG. 2 GROUP SINK REGISTER											
2C	SCRATCHPAD MOVE	0	0	1	0	REGISTER GROUP SOURCE OR SINK				REG. SELECT		MOVE DIR.	DPW 1/2	DOUBLE PAD WORD ADDRESS				MOVE DIR:	P←R	R←P					
																		1/2 DPW:	LEFT	RIGHT					
3C	4 BIT MANIPULATE	0	0	1	1	REGISTER GROUP 4 BIT SOURCE & SINK				REG SEL.	MANIPULATE VARIANTS			4-BIT MANIPULATE LITERALS				MANIP. VARIANTS:	SET	AND	OR	EOR	INC	INC TEST	DEC	DEC TEST
4C	BIT TEST REL. BRANCH FALSE	0	1	0	0	REGISTER GROUP 4 BIT SOURCE				REG SEL.	TEST BIT NUMBER		DSP SIGN	RELATIVE BRANCH DISPLACEMENT MAG.				DSP SIGN	+	−						
5C	BIT TEST REL BRANCH TRUE	0	1	0	1	REGISTER GROUP 4 BIT SOURCE				REG SEL.	TEST BIT NUMBER		DSP SIGN	RELATIVE BRANCH DISPLACEMENT MAG.				DSP SIGN	+	−						
6C	SKIP WHEN	0	1	1	0	REGISTER GROUP 4 BIT SOURCE & SINK				REG SEL.	SKIP TEST VARIANTS			4-BIT TEST MASK				SKIP TEST VARIANTS:	ANY CLR/	ALL CLR/	EQL CLR/	ALL CLR	ANY CLR/	ALL CLR/	EQL CLR/	ALL CLR
7C	READ/WRITE MEMORY	0	1	1	1	R/W VAR.	COUNT FA/FL VARIANTS			DATA REG CODE		TW SIGN		DATA TRANSFER WIDTH MAGNITUDE				R/W VARIANTS:	READ MOP	WRITE FAT						
																		COUNT VARIANTS:			FLT T	FAT FL↓ L	FAL FL?	FAL	FLL	FAL FL↓
8C	MOVE 8 BIT LITERAL	1	0	0	0	REGISTER GROUP REG. SEL. IS 2				ENTIRE 8 BITS OF 8-BIT LITERAL								REG SELECT:	X	Y						
																		TW SIGN:	+	−						
9C	MOVE 24 BIT LITERAL	1	0	0	1	REGISTER GROUP REG. SEL. IS 2				8 MOST SIGNIFICANT BITS OF FULL 24-BIT LITERAL															
10C	SHIFT/ROTATE T REGISTER	1	0	1	0	SINK REGISTER GROUP				SINK REG. SELECT		S/R VAR		LEFT SHIFT/ROTATE COUNT				S/R VARIANTS:	SHIFT	ROTATE						
11C	EXTRACT FROM T REGISTER	1	0	1	1	RIGHT BIT POINTER FOR EXTRACTION FIELD					SINK REG CODE			EXTRACTION FIELD WIDTH				SINK REG CODE:	X	Y	T	L				
12C	BRANCH REL FORWARD	1	1	0	0	RELATIVE DISPLACEMENT MAGNITUDE																			
13C	BRANCH REL. REVERSE	1	1	0	1	RELATIVE DISPLACEMENT MAGNITUDE																			
14C	CALL RELATIVE FORWARD	1	1	1	0	RELATIVE CALL ADDRESS MAGNITUDE																			
15C	CALL RELATIVE REVERSE	1	1	1	1	RELATIVE CALL ADDRESS MAGNITUDE																			
1D																									
2D	SWAP MEMORY	0	0	0	0	0	0	1	0	DATA REG CODE		TW SIGN		DATA TRANSFER WIDTH MAGNITUDE				TW SIGN:	+	−						
																		REG CODE:	X	Y	T	L				
3D	CLEAR REGISTER	0	0	0	0	0	0	1	1	L REG	T REG	Y REG	X REG	FA REG	FL REG	FU REG	CP REG								
4D	SHIFT/ROTATE X OR Y	0	0	0	0	0	1	0	0	S/R DIRECTION VARIANTS		X/Y VAR	LEFT OR RIGHT. X OR Y SHIFT/ROTATE COUNT					X/Y VAR:	X	Y						
																		S/R VAR:	SFT←	SFT→	ROT←	ROT→				
5D	SHIFT/ROTATE X AND Y	0	0	0	0	0	1	0	1	S/R DIRECTION VARIANTS		LEFT OR RIGHT X AND Y SHIFT/ROTATE COUNT						S/R DIR VAR:	SFT←	SFT→	ROT←	ROT→				
6D	COUNT FA/FL	0	0	0	0	0	1	1	0	COUNT FA/FL VARIANTS		COUNT SCALAR MAGNITUDE						COUNT FA/FL VAR:	NOP	FAT	FLL	FAL FL↓	FAL FLT	FAL	FLL	FAL FL↓
7D	EXCHANGE DPW	0	0	0	0	0	1	1	1	SINK DPW ADDRESS				SOURCE DPW ADDRESS											
8D	SCRATCHPAD RELATE FA	0	0	0	0	1	0	0	0				DSP SIGN	LEFT HALF PAD WORD ADDRESS				DSP SIGN:	+	−					
9D	MONITOR	0	0	0	0	1	0	0	1	LITERAL OCCURRENCE IDENTIFIER															
10D	NANO MOVE	0	0	0	0	1	0	1	0	C/A VAR	STOPPING SEQUENCE NUMBER					NANO-REG WORD PORTION		C/A VAR: SEQ NUMBER WORD PORTION	COUNTN 0/8 0	ABORT 1/9 1	2/10 2	3/11 3	4/12 4	5/13 5	6/14 6	7/15 7
11D	DIAG READ/WRITE MEMORY	0	0	0	0	1	0	1	1	DATA REG CODE	TW SIGN	REG.	M/E VAR.	R/W VAR	ECHO VAR.			DATA REG: R/W VAR: M/E VAR ECHO VAR TW SIGN:	X READ MEMORY RES +	Y WRITE ECHO WRITE −	T ADR	L ELR/CLR				
12D																									
13D																									
14D																									
15D																									

REGISTER SELECT

REGISTER GROUP	0	1	2	3
0	TA	FU	X	SUM
1	TB	FT	Y	CMPX
2	TC	FLC	T	CMPY
3	TD	FLD	L	XAMY
4	TE	FLE	A	XEQY
5	TF	FLF	M	MSKX
6	CA	BICN	BR	MSKY
7	CB	FLCN	LR	XORY
8	LA	NULL-A	FA	DIFF
9	LB	RESV	FB	MAXS
10	LC	PERM	FL	NULL-C
11	LD	PERP	TAS	U
12	LE	XYCN	CP	NULL-D
13	LF	XYST	NULL-B	DATA
14	CC	INCN	CSW	CMAD
15	CD	MISW	PZMA	NULL

NOTE: BITS ARE NUMBERED HERE ACCORDING TO THE HARDWARE CONVENTIONS:



INDIVIDUAL BITS OF SOME 4-BIT REGISTERS HAVE SPECIAL MEANINGS AS NOTED BELOW:

	3	2	1	0
BICN	LSUY	CYF	CYD	CYL
XYCN	MSBX	X=Y	X<Y	X>Y
XYST	LSUX	INT	Y NEQ 0	X NEQ 0
FLCN	FL=SFL	FL>SFL	FL<SFL	FL NEQ 0
INCN	PORT DEVICE MISSING	PORT HI PRIORITY	PORT INTERRUPT	PORT LOCKOUT
CC	CONTROL FNL STATE LAMP FLIP-FLOP	REAL TIME CLOCK INTERRUPT	I/O BUS SERV. REQST INTERRUPT	CONTROL FNL INTERRUPT
CD	MEM READ ERROR INTERRUPT	MEM WRITE/SWAP (LR/BR CHK) OUT OF BDS OVERRIDE	MEM READ ADDR (LR/BR CHK) OUT OF BDS	MEM WRITE/SWAP (LR/BR CHK) OUT OF BDS INTERRUPT
PERP	CACHE DOUBLE HIT	CACHE KEY PARITY ERROR ON KEY A OR KEY B	PARITY ERROR OF FETCHED MICRO IN M-REGISTER	CASSETTE ERROR NOT CORRECTABLE
PERM	S-MEMORY MICRO INSTRUC. ON TIME-OUT	S-MEM FIELDS OUT OF ADMINISTRATIVE BOUNDS	S-MEMORY E.CC. REG. LOADED	UNCORRECTABLE S-MEMORY ERROR, SEE PROC 073
MISW	NULL BIT	NULL B.T	32 MICRO SOURCE FROM CACHE OR PROGRAM M. REGISTER	32 MICRO SOURCE FROM S-MEM OR PROGRAM M. REGISTER

INT = ANY ONE OR MORE OF: CC0, CC1, CC2, CD0, CD3, INCN 1, INCN 3.

	MICRO NAME	MC 15	14	13	12	MD 11	10	9	8	ME 7	6	5	4	MF 3	2	1	0
1E	DISPATCH	0	0	0	0	0	0	0	0	0	0	0	1	DISPATCH VARIANTS			SKIP FLAG
2E	CASSETTE CONTROL	0	0	0	0	0	0	0	0	0	0	1	0	CASSETTE MANIP VARIANTS			MARK 0 SKIP=1
3E	BIAS	0	0	0	0	0	0	0	0	0	0	1	1	BIAS VARIANTS			TEST FLAG
4E	STORE F INTO DPW	0	0	0	0	0	0	0	0	0	1	0	0	SINK DPW ADDRESS			
5E	LOAD F FROM DPW	0	0	0	0	0	0	0	0	0	1	0	1	SOURCE DPW ADDRESS			
6E	CARRY FF MANIPULATE	0	0	0	0	0	0	0	0	0	1	1	0	CYF ← CYD	CYF ← CYL	CYF ← 1	CYF ← 0
7E	READ/WRITE CACHE	0	0	0	0	0	0	0	0	0	1	1	1	BD PRTY GD PRTY	READ/WRITE VARIANTS		
8E																	
9E																	
10E																	
11E																	
12E																	
13E																	
14E																	
15E																	
1F	HALT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2F																	
3F	NORMALIZE X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4F	BIND	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5F	CLEAR CACHE	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
6F																	
7F																	
8F	INC A	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
9F																	
10F																	
11F																	
12F																	
13F																	
14F																	
15F																	
ZERO	NO OPERATION	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VARIANTS:	0 00 000	0 01 001	10 010	11 011	100	101	110	111
SKIP FLAG: DISP VAR:	FAIL LOCK	SUCC WRTLO	READ	R & C	WRTHI	ABSNT	UNDEF	UNDEF
CASSETTE MANIP VAR:	START TAPE	STOP ① GAP	STOP ON X≠Y	STOP ON FA≠BR	UNDEF	UNDEF	STOP ON X=Y	STOP ON FA=BR
TEST FLAG: BIAS VAR:	TEST/ UNIT	TEST F	S	FS	NO-OP	FCP	NO-OP	NO-OP
READ/WRITE VAR:	RESV.	RESV.	RESV.	RESV.	RESV.	DIAG WRITE	READ MICRO	READ KEYS
PARITY:	GOOD	BAD						

NOTE:

THE ERROR LOG REGISTER HAS MEANINGS AS FOLLOWS:

M	PU	MU	S	W	BOARD ROW	SYNDROME

PU: UNCORRECTABLE ERROR ON CPU ACCESS

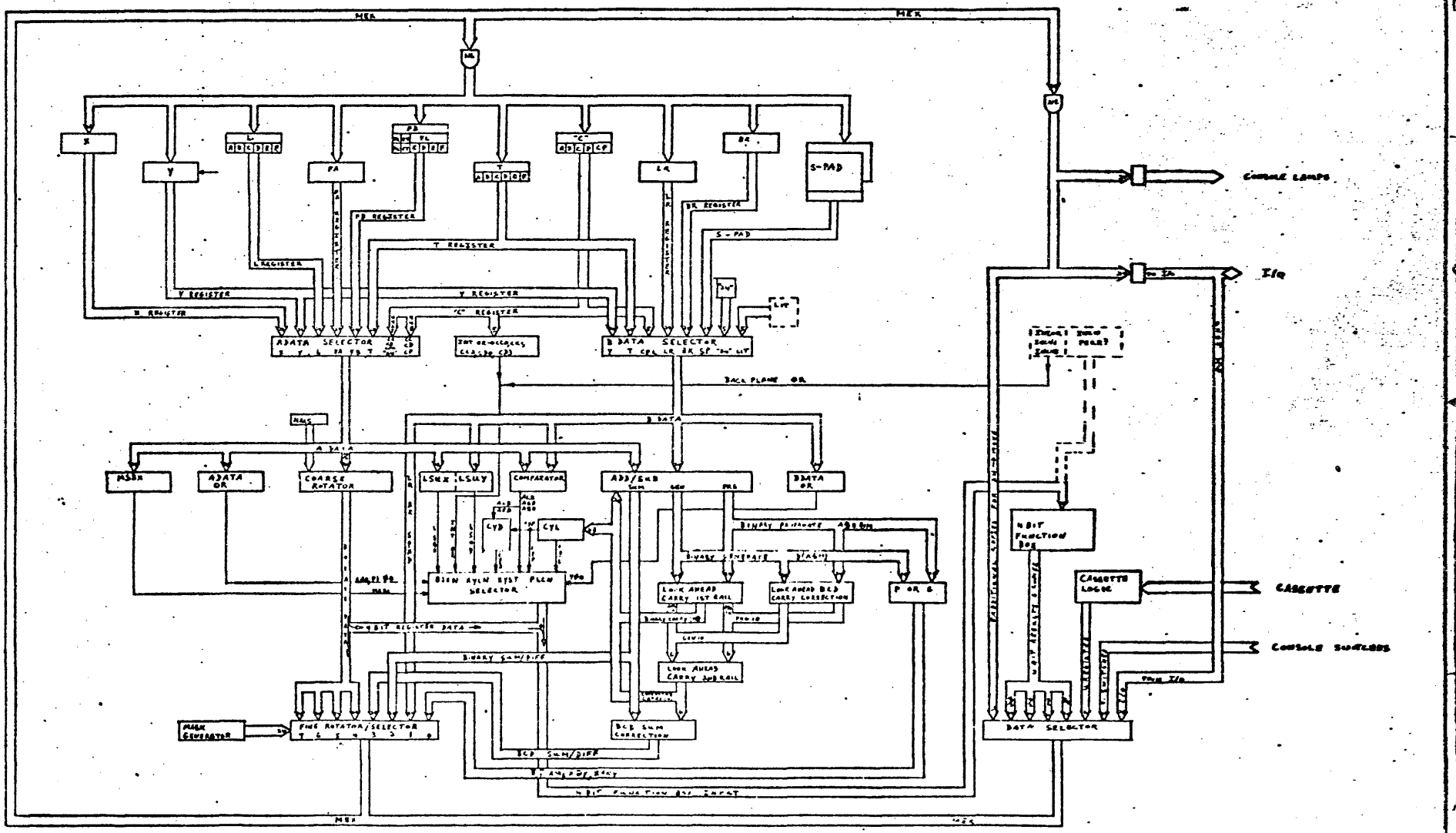
MU: UNCORRECTABLE ERROR ON NON-CPU ACCESS

S: SINGLE BIT ERROR (CORRECTED IN DATA OR CHECK-BIT)

M: MULTIPLE ERROR OCCURRENCE (LOGGING DATA LAST)

W: BOARD/ROW/SYNDROME LOADED ON WRITE OPERATION

Fig 3 BLOCK DIAGRAM EXECUTE STRUCTURE



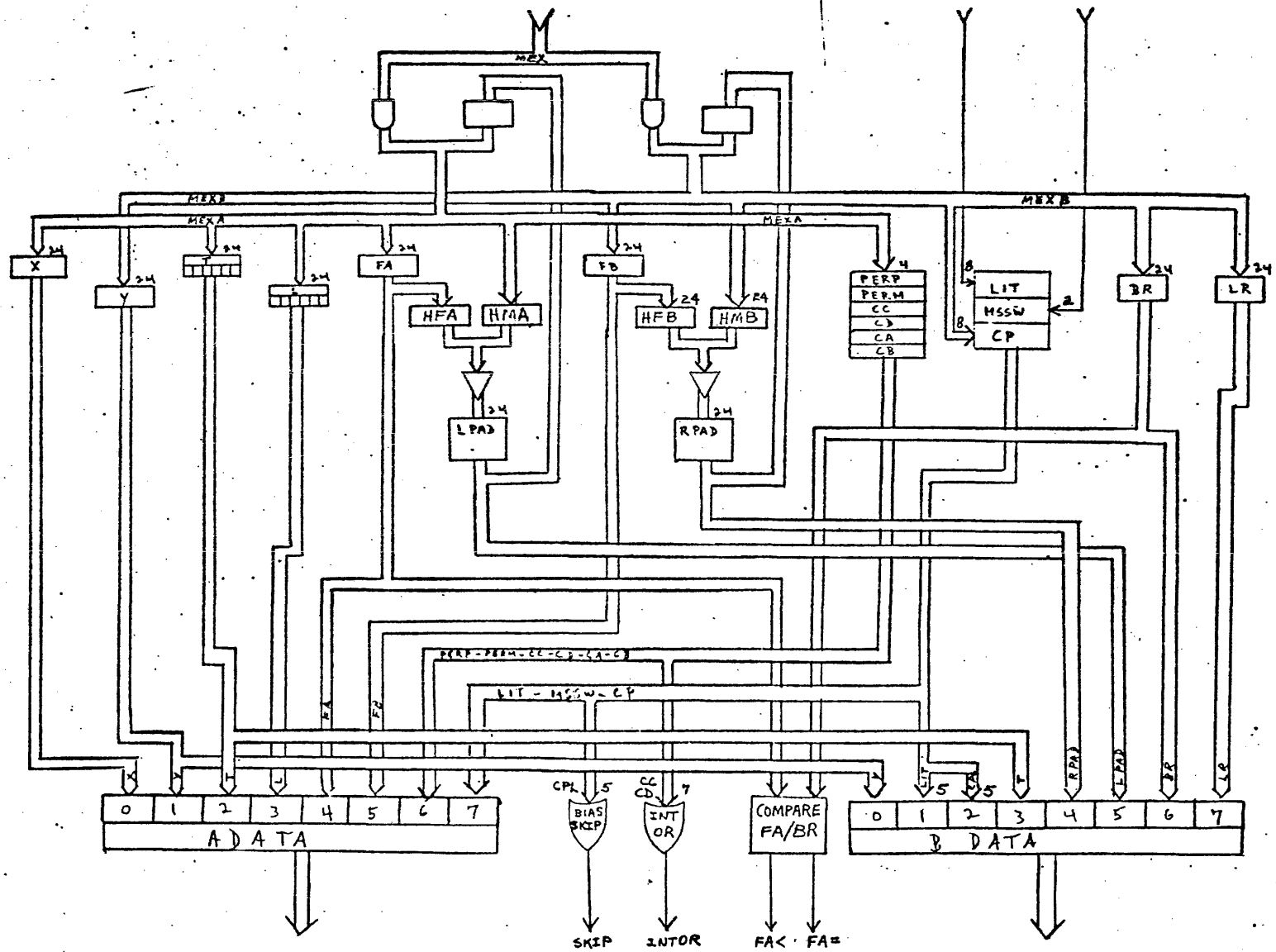


Fig 4A

BLOCK DIA.
B1820 CARDS
H3, K3 & M3
2/11/76
F. Williams

A DATA SELECTORS						
ASELn.X0 n=9,2	CARD H (23 … 16)		CARD K (15 … 8)		CARD M (7 … 0)	
0	X	X	X	X	X	X
1	Y	Y	Y	Y	Y	Y
2	TA	TB	TC	TD	TE	TF
3	LA	LB	LC	LD	LE	LF
4	FA	FA	FA	FA	FA	FA
5	FU	FT	FLC	FLD	FLE	FLF
6	PERP	PERM	CC	CD	CA	CB
7	8 LIT	8LIT	MSSW	—	CP	CP
	EA2316X0		EA1508X0		EA0700X0	

B DATA SELECTORS						
BSELn.X0 n=9,2	CARD H		CARD K		CARD M	
0	Y	Y	Y	Y	Y	Y
1	—	—	—	—	5LIT	5LIT
2	—	—	—	—	CPL	CPL
3	T	T	T	T	T	T
4	LPAD	LPAD	LPAD	LPAD	LPAD	LPAD
5	RPAD	RPAD	RPAD	RPAD	RPAD	RPAD
6	BR	BR	BR	BR	BR	BR
7	LR	LR	LR	LR	LR	LR
	EB2316X0		EB1500X0			

A DATA & B DATA SELECTORS
ASSIGNMENTS
W.K. Spruce 2/8/76

Fig 4B

V	NO.	DRAWN	CK.	APPD	DATE

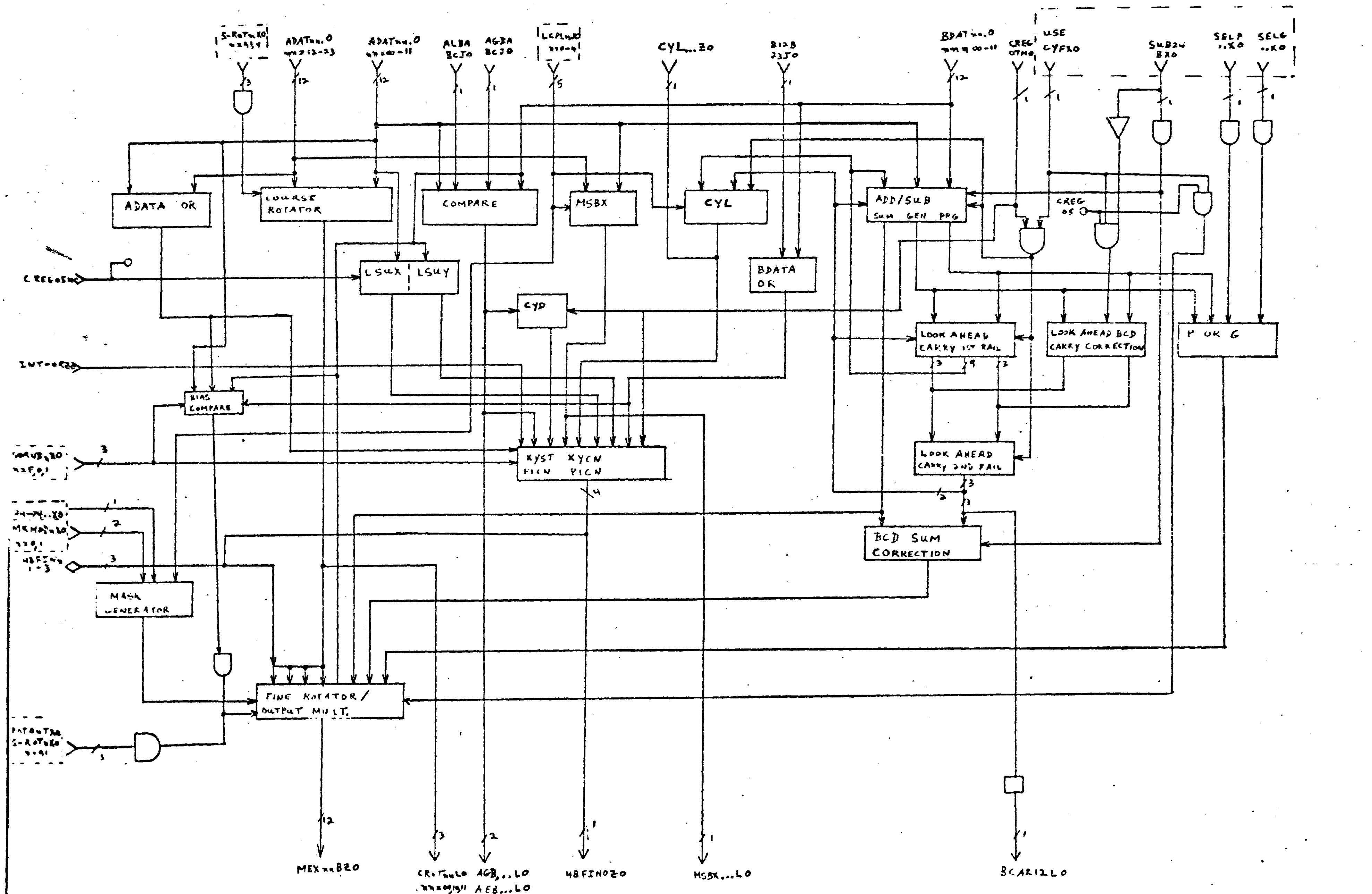


Fig 5 ALU SUBSTRUCTURE (12 LSB)

DRAWN W.K. SPROUSE	PROJ. W.K. Sprouse 2/2/76	**Burroughs Corporation**	TITLE BLOCK DIAGRAM – FUNCTION/ROTATOR LOWER HALF	PAGE 1/1
CHKD. Matsuda	APPD. mua 2 Feb 76	BUSINESS MACHINES GROUP GOLETA, CALIFORNIA 93017 — SMALL SYSTEMS PLANT U. S. AMERICA	UNIT M/PROC-3 L3	C-2216 0352 — A REV
PROPRIETARY TO BURROUGHS CORPORATION — NOT TO BE REPRODUCED NOR USED FOR MANUFACTURING PURPOSES, EXCEPT ON BURROUGHS ORDER OR PRIOR WRITTEN CONSENT.			SYSTEM B1820	

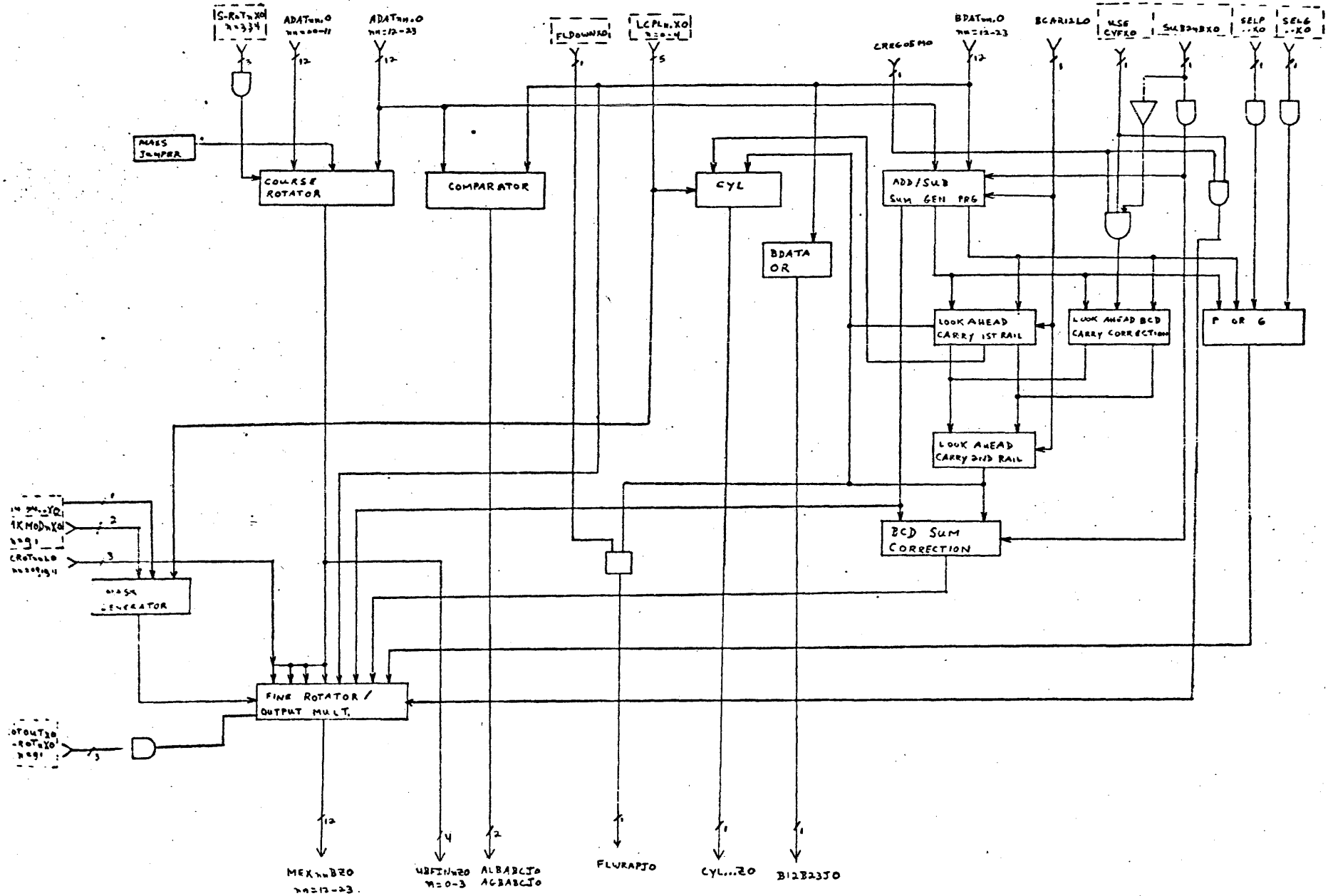


Fig 6. ALU SUBSTRUCTURE (12 MSB)

DRAWN W.K. SPROUSE	PROJ. W.K. Sprouse 2/2/76	Burroughs Corporation	TITLE BLOCK DIAGRAM - 2 OF 3 - F FUNCTION/ROTATER UPPER HALF
CHKD. R Montoya	APPD. W.K. Sprouse 2 Feb 76	BUSINESS MACHINES GROUP GOLETA, CALIFORNIA 93017 — SMALL SYSTEMS PLANT U. S. AMERICA	PAGE 1/1
PROPRIETARY TO BURROUGHS CORPORATION — NOT TO BE REPRODUCED NOR USED FOR MANUFACTURING PURPOSES, EXCEPT ON BURROUGHS ORDER OR PRIOR WRITTEN CONSENT.			UNIT M/PROC CD J3 — SYSTEM B1920 — C-2216 0345 — A REV

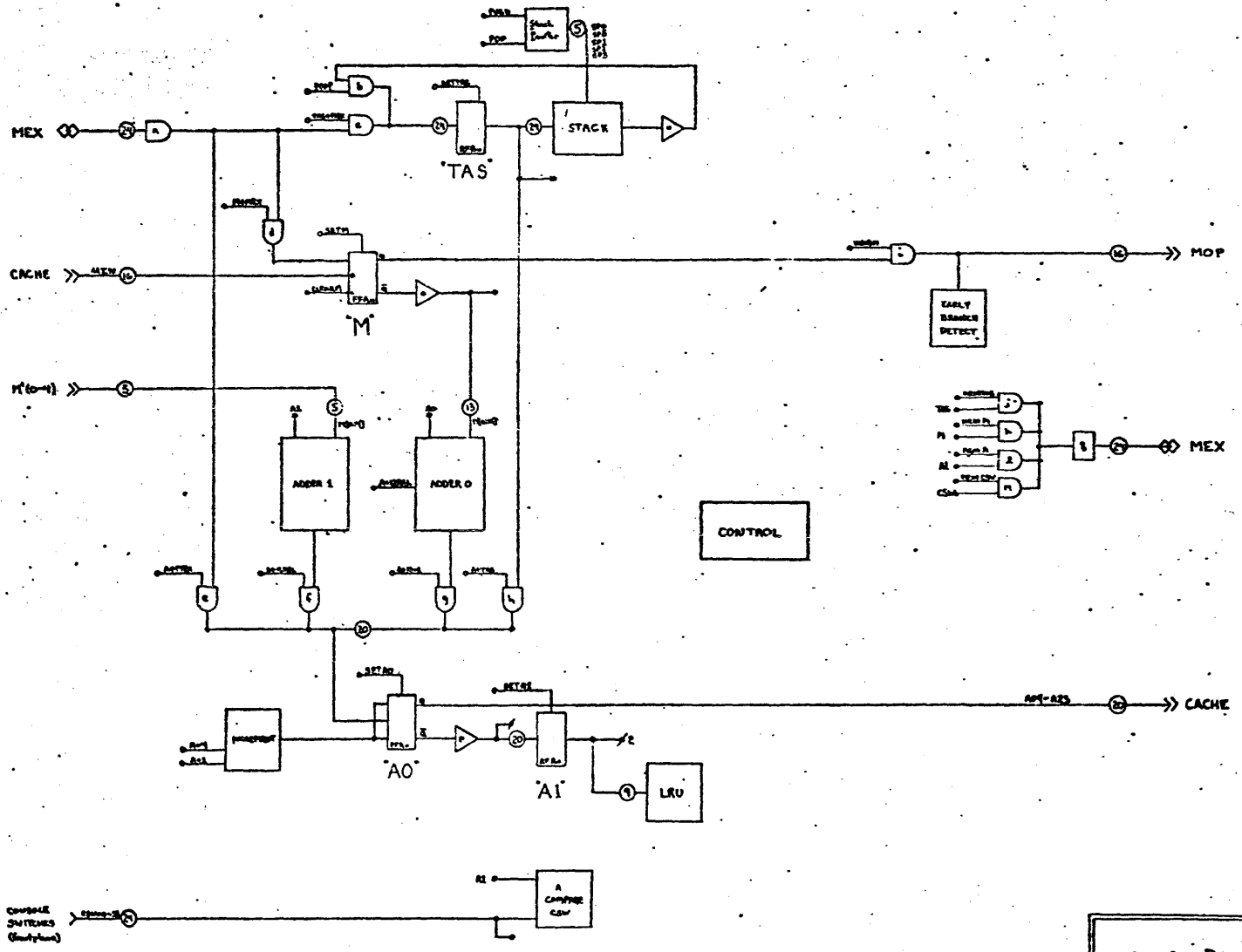


Fig 7 BASIC DATA PATHS FOR A, TAS, & M

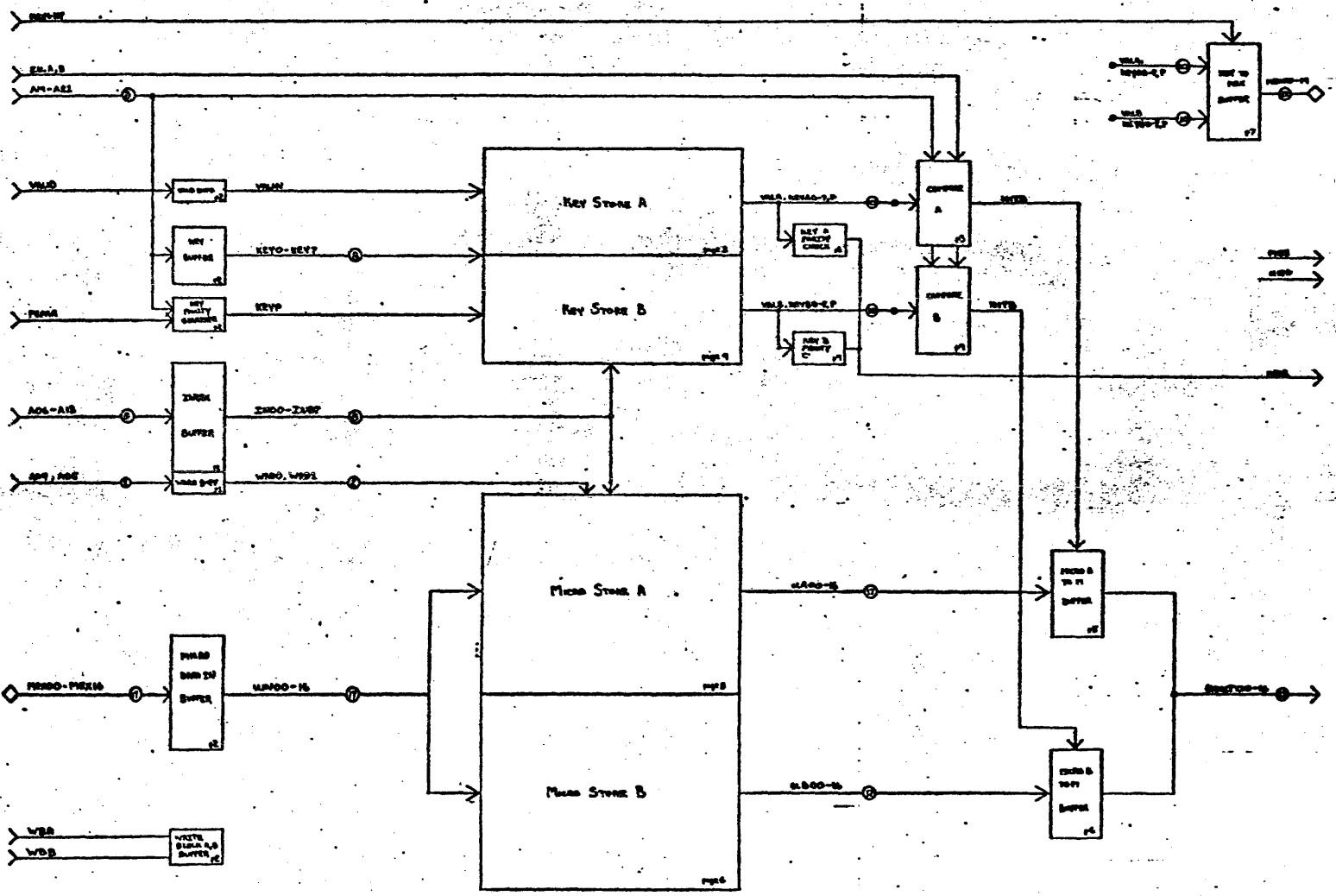


Fig 8. Block Diagram M-Proc-3 Cache

FIG 8

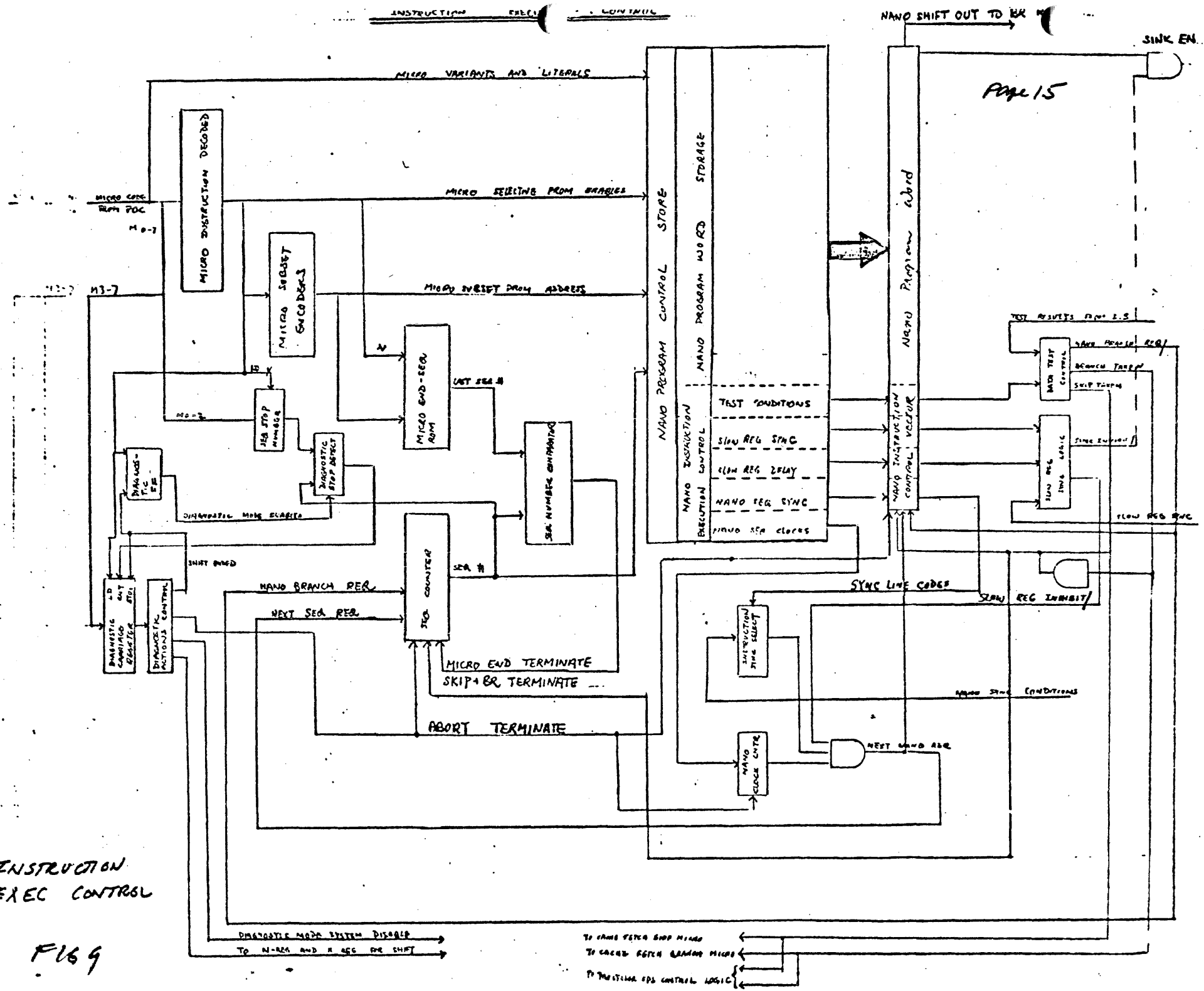

INSTRUCTION EXEC CONTROL
NANO SHIFT OUT TO BR
SINK EN.
Page 15
MICRO VARIANTS AND LITERALS
MICRO DESTINATION DECODED
MICRO SELECTING FROM BRANCHES
MICRO SUBSET ENCODER
MICRO SUBSET FROM ADDRESS
MICRO END-SEQ ROM
LAST SEQ #
JSB STOP MICRO
DIAGNOSTIC STOP DETECT
SEQ NUMBER COMPARING
DIAGNOSTIC MODE ENABLES
SHIFT ENCD
DIAGNOSTIC CARRIAGE REGISTER
DIAGNOSTIC ACTIONS CONTROL
NANO BRANCH REG
NEXT SEQ REG
SEQ COUNTER
SEQ #
MICRO END TERMINATE
SKIP+BR TERMINATE
ABORT TERMINATE
NANO PROGRAM CONTROL STORE
NANO PROGRAM WORD STORAGE
NANO INSTRUCTION EXECUTION CONTROL
TEST CONDITIONS
SINK REG SYNC
FLOW REG DELAY
NANO REG SYNC
MICRO SEQ CLOCKS
NANO Program word
NANO INSTRUCTION CONTROL VECTOR
TEST RESULTS FROM E.S
BRANCH TAKEN
SKIP TAKEN
SINK REG INHIBIT
FLOW REG SYNC
SYNC LINE CODES
INSTRUCTION SYNC SELECT
NANO CLOCK CNTR
NEXT NANO ADR
DIAGNOSTIC MODE SYSTEM DISABLE
TO N-REG AND R REG RE SHIFT

FIG 9

2.1 THREE PHASE MICRO EXECUTION

Referring to Figure 10, the three phase execution of a micro is described below.

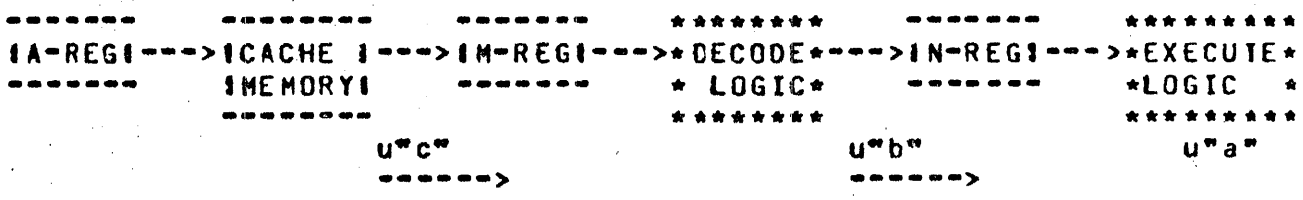


FIGURE 10

The process of a micro-instruction in the machine has three steps. Assume a string of one clock micros. At the first clock, it is loaded into the instruction register (M); at the second, it is decoded and stored in the control register (N); and at the last clock, it is executed. At one clock time, all three steps are occurring, each to a different micro-instruction. Stated differently, while the fetch of the micro-instruction (c) one beyond the next micro-instruction is being made to the M-register, the next micro-instruction (b) is being decoded into the N-register, and the present micro-instruction (a) is being executed. This micro-instruction process will be referred to as the 3-phased cycle.

Figures 11 through 20 show stylized 3-phase execution wavefcrms.

Figures 21 and 22 show a breakdown of the N-register; Figure 21 the low order bits and Figure 22 the high order bits.

CLOCK	T1	T2	T3	T4	T5
N-REG	M1(1)	M2(1)	M3(1)*	M3(1)*	M4
M-REG	M2	M3	M4	M4	M5
FETCH	M3	M4	M5	M5	M6
A1	a2	a3	a4	a4	a5
A0	a3	a4	a5	a5	a6

A1 = INSTRUCTION REG (address of instruction in M-REG, being decoded)

A0 = CACHE ADDRESS REG (address of instruction being fetched from Cache

aJ = Address of MJ

MJ = Micro-instructions J

MJ(i) = ith nano instruction, of micro-instruction J

*MJ(i) held over for another clock

MJ --> MJ(i) : Micro J being decoded into its ith nano

FIGURE 11 SINGLE NANO MICRO-INSTRUCTION TIMINGS

N	T1	T2	T3	T4	T5	
N-REG	M2	M2(1)	M2(2)	M2(3)	M3	
M-REG	M2->M2(1)	M2->M2(2)	M2->M2(3)	M3	M4	
FETCH	M3	M3	M3	M4	M5	
A1	a2	a2	a2	a3	a4	
A0	a3	a3	a3	a4	a5	

FIGURE 12 MULTI-NANO TIMINGS (3 NANO INSTRUCTION)

Note: Figures 13 thru 22 are missing.

3.0 PRODUCT DESCRIPTION

3.1 PROCESSOR REGISTERS

3.1.1 M

The M-register (micro-register) is a 16-bit register used tc hold the active micro-instruction (M-instruction or micrc-operator or M-op). The state of this register is decoded to enable the cifferent control signals within the processor to perform the operations called for by the M-Op. The M-register is broken into 4 fields for decoding that are structured such that 61 distinct M-ops can be decodec.

The M-register is addressable as a source and as a sink (destination). When used as a sink register, the source is bit-ORed with the upcoming M-op. Exception: In TAPE mode, the source is not bit-ORed with the upcoming M-op.

There is a 17th bit for parity associated with the M-register. It contains the odd parity during a fetch from the cache or S-memory (but not the cassette) to the M-register. A parity check is performec on the contents of the M-register and the parity bit after each fetch. The exceptions of the parity check are after a a cassette lcad, a console load, or a move to M-register (bit OR with the next micro.)

3.1.2 A-REGISTER

The A-register is a 18-bit micro-program address register capable of addressing 262,144 (256KB) micro-operators located in cache and/or S-memory.

The A-register is capable of having 12 bit binary increments with values from 0 through 4095 addec to or subtracted from it. A high-speed carry adder facilitates micro-program branching. The A-register is automatically incremented during RUN mode. RUN mode includes CONTINUOUS, OR STEP. In TAPE, the A-register is not automatically incremented by 1. However, any SKIP or BRANCH will result in the A-register being modified accordingly. Wrap-arcund can occur and is permitted.

The A-register can be addressed as a source and as a sink. The A-register with 18 bits lines up with the main exchange (MEX) from MEX04 to MEX21. When used as a source, the contents of the A-register are multiplied by 16. When used as a destination, the rightmost 4 bits of the source are lost.

To obtain the micro-operator from S-memory, the A address is multiplied by 16 to yield a bit address pointing tc the micro-operator. To obtain the micro-operator from cache, the A-address points directly to the cache location. A micro-operator is obtained if the cache is enabled and if the micro-operator resides in cache (i.e., the validity bit indicates a micro is present and there is a "HIT").

The A-register is addressed as destination register by the micro-operator "BIND".

3.1.3 TAS

The TAS (Top of Stack) Register is a 24-bit register which is the top of the A-Stack. The TAS register is addressable as a source and as a sink. References to TAS result in the stack being automatically pushed or popped.

3.1.4 A-STACK

The A-Stack is a 33 word deep 24-bit wide memory, without automatic hard overflow interrupt, which operates as a push-down stack with a last-in, first-out type of structure. The TAS register operates in conjunction with the A-Stack to provide a virtual stack of 33 words deep. Using this stack, the nano-routines operate in the normal software call-return type of programming. This allows for a highly shared nano-structure and reduces the nano-memory requirements. Whenever the nano-routine uses the stack, it only uses one location of the stack. After each micro, any nano push of the stack is always completed with a nano pop of the stack. The user is only guaranteed 32 locations of the A-stack, since one location must be available for the nano-program.

The stack pointer has 32 states. Since the physical A-stack has 33 locations, care must be exercised when the A-stack is used for more than 32 consecutive pushes or pops. As a note of caution, the first pop after a push is a destructive read which overwrites the TAS with the next word.

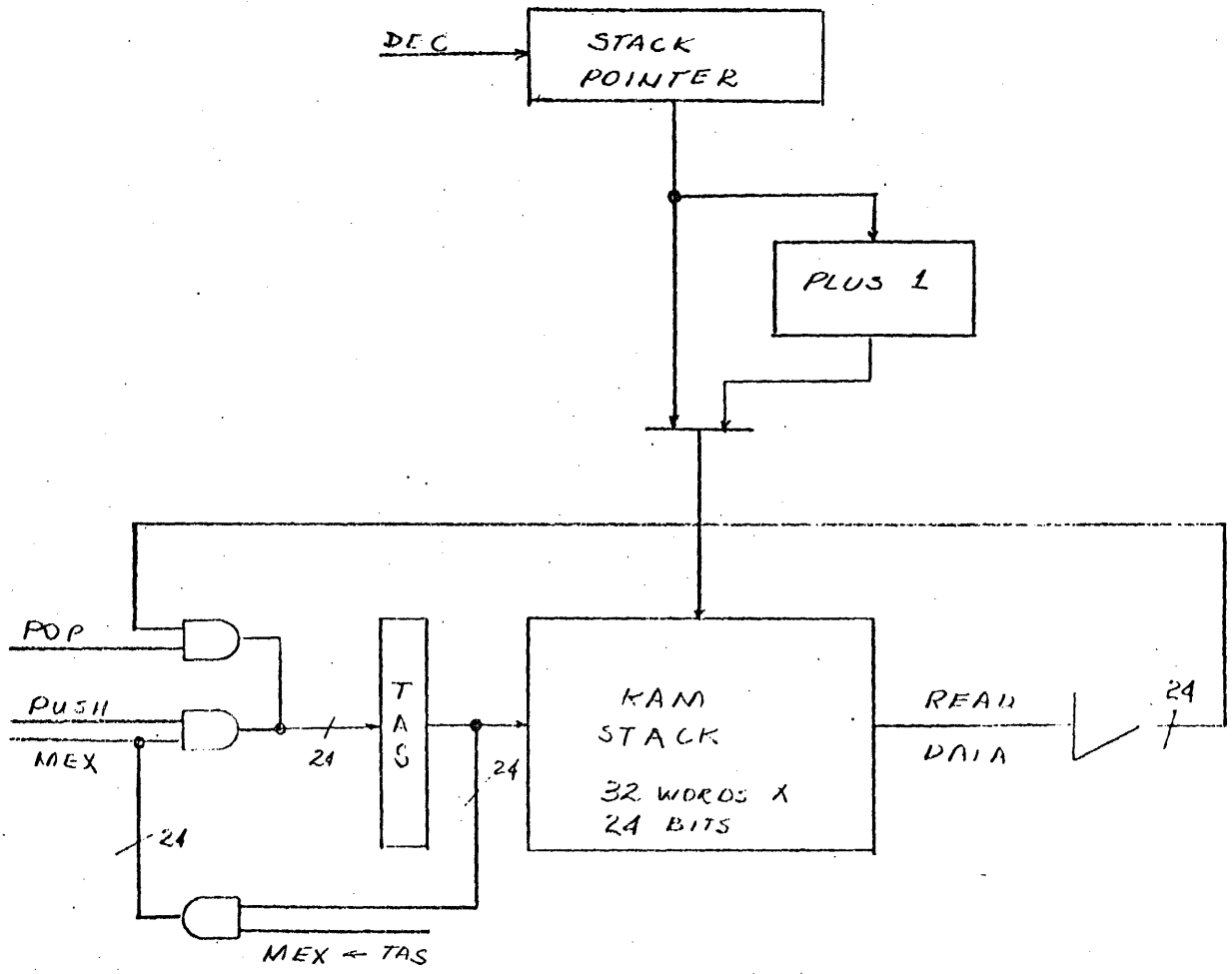


FIGURE 23 BLOCK DIAGRAM OF A-STACK

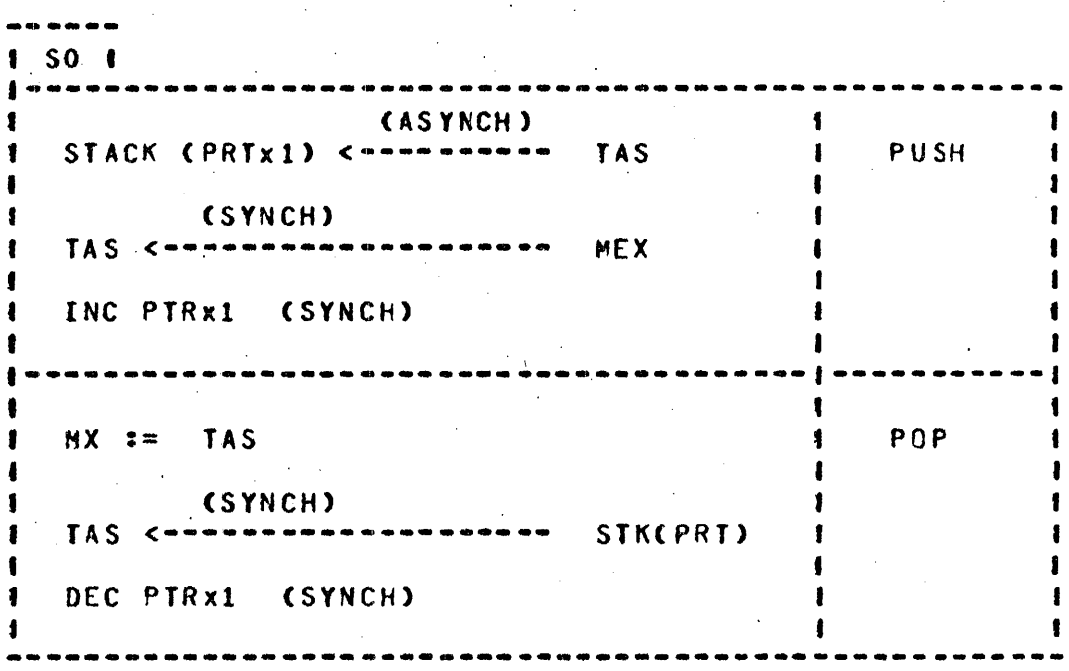


FIGURE 24 FLOW OF A-STACK OPERATION

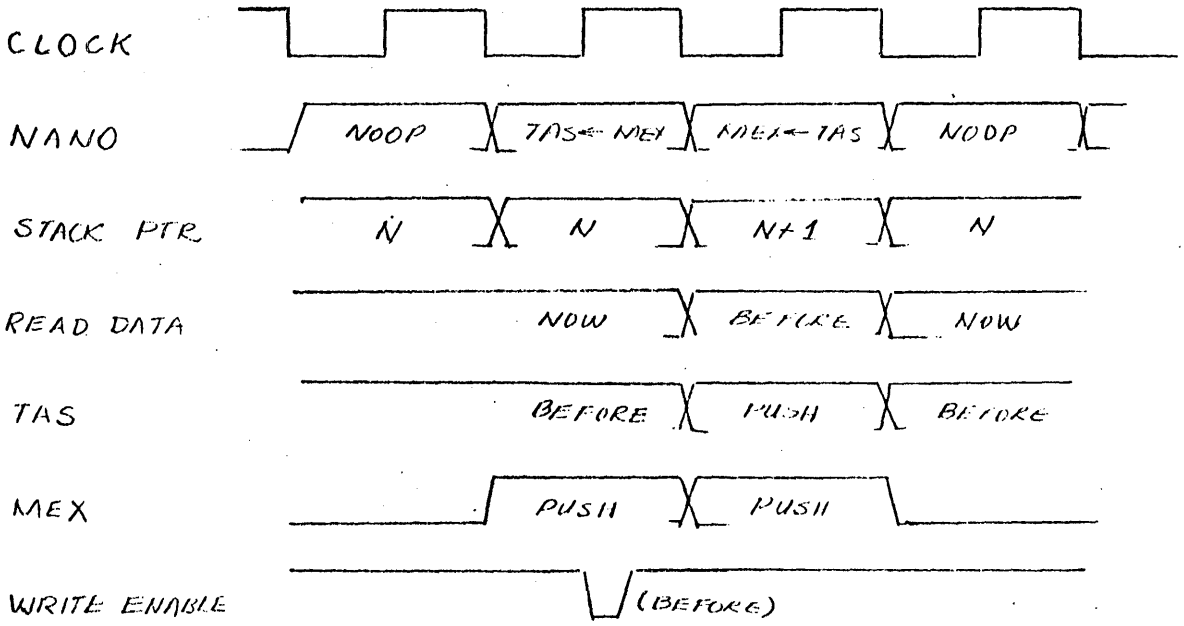


FIGURE 25 TIMING OF A-STACK OPERATION

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

M-PROCESSOR-3
E.D.S. #2215 9891

	TAS	0	1	2	3	4	.....	27	28	29	30	31	PRT	RAM BEHAVIOR	
CLEAR	0	0	0	0	0	0		0	0	0	0	0	0		
PUSH 1	1	0	(0)	0	0	0		0	0	0	0	0	1	1st WRITE	@1
PUSH 2	2	0	0	(1)	0	0		0	0	0	0	0	2	2nd WRITE	@2
PUSH 3	3	0	0	1	(2)	3		0	0	0	0	0	3	3rd X	@3
PUSH 4	4	0	0	1	2	(3)		0	0	0	0	0	4	4th X	@4
•															
•															
PUSH 27	27	0	0	1	2	3		(26)	0	0	0	0	27	27th WRITE	@27
PUSH 28	28	0	0	1	2	3		26	(27)	0	0	0	28	28th X	@28
PUSH 29	29	0	0	1	2	3		26	27	(28)	0	0	29	29th X	@29
PUSH 30	30	0	0	1	2	3		26	27	28	(29)	0	30	30th X	@30
PUSH 31	31	0	0	1	2	3		26	27	28	29	(30)	31	31st X	@31
PUSH 32	31	(31)	0	1	2	3		26	27	28	29	30	0	32nd X	@ 0
PUSH 33	33	31	(32)	1	2	3		26	27	28	29	30	1	1st OVERWRITE	@1
PUSH 34	34	31	32	(33)	2	3		26	27	28	29	30	2	2nd OVERWRITE	@2
PUSH 35	35	31	32	33	(34)	3		26	27	28	29	30	3	3rd X	@3
PUSH 36	36	31	32	33	34	(35)		26	27	28	29	30	4	4th X	@4
•													•		
PUSH 59	59	31	32	33	34	35		(58)	27	28	29	30	27	27th OVERWRITE	@27
PUSH 60	60	31	32	33	34	35		58	(59)	28	29	30	28	28th X	@28
PUSH 61	61	31	32	33	34	35		58	59	(60)	29	30	29	29th X	@29
PUSH 62	62	31	32	33	34	35		58	59	60	(61)	30	30	30th X	@30
PUSH 63	63	31	32	33	34	35		58	59	60	61	(62)	31	31st X	@31
PUSH 64	64	63	32	33	34	35		58	59	60	61	62	0	32nd X	@0
PUSH 65	65	63	64	33	34	35		58	59	60	61	62	1	33rd X	@1
PUSH 66	66	63	64	65	34	35		58	59	60	61	62	2	34th X	@2
PUSH 67	67	63	64	65	66	35		58	59	60	61	62	3	35th X	@3

FIGURE 26 MULTIPLE PUSHES

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
M-PROCESSOR-3
E.D.S. #2215 9891

		TAS	C	1	2	3	4	...	27	28	29	30	31	PRT	DATA OUT TO MEX
PUSH	33	33	31	(32)	1	2	3		26	27	28	29	30	1	--
POP	1	32	31	[32]	1	2	3		26	27	28	29	30	0	33
POP	2	31	[31]	32	1	2	3		26	27	28	29	30	31	32
POP	3	30	31	32	1	2	3		26	27	28	29	[30]	30	31
POP	4	29	31	32	1	2	3		26	27	28	[29]	30	29	30
POP	5	28	31	32	1	2	3		26	27	[28]	29	30	28	29
.														.	.
.														.	.
POP	30	3	31	32	1	2	[3]		26	27	28	29	30	3	4
POP	31	2	31	32	1	[2]	3		26	27	28	29	30	3	3
POP	32	1	31	32	[1]	2	3		26	27	28	29	30	1	2
POP	33	32	31	[32]	1	2	3		26	27	28	29	30	0	1
POP	34	31	[31]	32	1	2	3		26	27	28	29	30	31	32
POP	35	30	31	32	1	2	3		26	27	28	29	[30]	30	31
POP	36	29	31	32	1	2	3		26	27	28	[29]	30	29	29
.														.	.
POP	64	1	31	32	[1]	2	3		26	27	28	29	30	1	2
POP	65	32	31	[32]	1	2	3		26	27	28	29	30	0	1
POP	66	31	[31]	32	1	1	3		26	27	28	29	30	31	32
POP	67	30	31	32	1	2	3		26	27	28	29	30	30	31
POP	68	29	31	32	1	2	3		26	27	28	29	30	29	30

FIGURE 27 MUTIPLE POPS

3.1.5 X AND Y

The X-register and the Y-register are 24-bit general purpose registers used primarily to hold and to act as a source for the two operands of the arithmetic and combinatorial unit. Each register is addressable as a source and as a sink.

Both registers are capable of all SHIFT/ROTATE operations. The X-register is capable of the NORMALIZE. For cache diagnostic READ/WRITE operation, the X-register is used to hold data. For the cache diagnostic WRITE, the Y-register is also used for data.

The X and Y-registers are two of the four registers (X, Y, T, and L) capable of the diagnostic cache write operation with the cache memory.

The Y-register is the destination on the Diagnostic Read/Write Memory (11D) operation for the echo response of the write data and address, and the ELOG register.

The X- and Y- registers are compared in the cassette control micro to either cause a halt or a skip depending upon the variant (see section 3.4.30).

3.1.6 L

The L-register is a 24-bit general purpose register used typically to hold logical flags for the micro-program code. The L-register, as well as each 4-bit group of L (denoted as LA, LB, LC, LD, LE, and LF), is addressable as a source and as a sink.

DISPATCH operations use the L-register as the source or sink for a 24-bit message (usually an address) which is stored in/fetched from S-memory location zero.

The BIND operator uses the L-register as the source of the 24-bit value to be added to T as a sum going to A.

Since the L-register is addressable in 4-bit groups, its contents are available for analysis and alteration via the 4-bit function box. MANIPULATE, SKIP and BIT-TEST-BRANCH instructions can operate on L data.

The L-register is one of 4 registers (X, Y, L, and T) capable of read/write operations with main memory.

The L-register is one of the 4 registers (X, Y, T and L) capable of the diagnostic cache write operation with the cache memory.

3.1.7 T

The T-register is a 24-bit general purpose register used primarily for the interpretation of the S-language instructions. T-register, as well as each 4-bit group of T (denoted as TA, TB, TC, TD, TE, and TF), is addressable as a source and as a sink.

DISPATCH operations use the least significant seven bits of T as the source or sink for the port and channel information associated with the DISPATCH operation.

The BIND operator moves (L plus T) to A.

Since the T-register is addressable in 4-bit groups, its contents are available for analysis and alteration via the 4-bit function box. MANIPULATE, SKIP and BIT-TEST-BRANCH instructions can operate on T data.

The T-register is one of 4 registers (X, Y, L, and T) capable of both read/write operations with main memory and of the diagnostic cache write operation with the cache memory.

The T-register is also capable of the SHIFT/ROTATE and EXTRACT operations.

3.1.8 FA

The FA-register (Field Address) is a 24-bit register used primarily to hold an absolute bit address for the main memory. It has the capability of directly addressing any bit in the memory starting at any point. The FA-register is addressable as a source and as a sink.

The FA-register is capable of being counted up or down by a literal in a micro-instruction or by the value contained in CPL in order to facilitate iteration through a memory field. It can be incremented or decremented by a value in a left scratchpad word. It also has the capability of being loaded, stored, or swapped along with FB into a double scratchpad word.

Neither overflow nor underflow of FA is detected. The value of FA may go through its maximum value or its minimum value and wrap around.

The READ/WRITE Cache diagnostic operator uses the FA-register as the source of the cache address to be used in the operation.

The FA-register can be compared with the BR-register for three functions. In the memory operations, FA is checked to be within the limits of the BR-register (see Sections 3.1.12 or 3.1.13.2), or it can be used as a halt variant in the cassette control micro, or it can be used as a SKIP variant in the cassette control micro. (see Section 3.4.30).

3.1.9 FB

The FB-register is a 24-bit register which can be functionally divided into three portions: a 4-bit FU (Field Unit) register, a 4-bit FT (Field Type) register, and a 16-bit FL (Field Length) register.

The FB-register, as well as each 4-bit portion of FB denoted as FU, FT, FLC, FLD, FLE, and FLF is addressable as a source and as a sink. In addition, the 16-bit portion comprised of FLC, FLD, FLE, and FLF and denoted as FL is also addressable as source and as a sink.

The FU-register holds the length of the unit which makes up a field in memory. The FT-register holds the field tape information while the FL-register holds the total length of the field. FL is capable of describing fields up to 65,636 bits and can be adjusted up or down by a literal in a micro-instruction or by the value contained in CPL in order to facilitate iteration through a memory field.

Note: Overflow of FL is not detected. The value of FL will go through its maximum value and wrap around. Underflow of FL is detected and will not wrap around. The value of zero is left in FL.

Since the FB-register is addressable in 4-bit groups, its contents are available for analysis and alteration via the 4-bit function box. MANIPULATE, SKIP and BIT-TEST-BRANCH instructions can operate on FB data.

FB has the capability of being loaded, stored, or swapped along with FA into a double scratchpad memory word.

FU and FL along with corresponding portions of the first cell of scratchpad are used to set the various conditions of FLCN (see Section 3.1.11, FLCN) and the various conditions of the CP-register (see Section 3.4.20, BIAS).

3.1.10 SCRATCHPAD

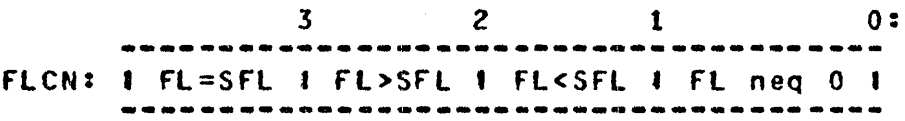
A scratchpad memory of sixteen 48-bit words is provided to hold field descriptors during the iteration of operands. Some cells may be used to hold S-language stack pointers and other processor registers which are under constant manipulation.

The FU and FL portion of the FB-register and like portions of the first cell of the scratchpad are used to set the various conditions of FLCN (see Section 3.1.11, FLCN) and the various conditions of the CP-register (see Section 3.4.20, BIAS).

The WRITE into scratchpad is a two step operation, each requiring one clock to execute. At the first clock, the data is trapped into a latch and during the second clock the data is written into the scratchpad. The second clock may occur in the micro execution time of another micro which did not direct this WRITE operation. If that micro was a READ from scratchpad, then it must be delayed until the previous WRITE operation was completed.

3.1.11 FLCN

FLCN (field length condition) is a 4-bit psuedo register that holds the result of a comparison of the FL portion of the FB-register and the corresponding portion of the first scratchpad word. It is addressable as a source only.



3.1.12 BR AND LR

The LR and BR-registers are both 24-bit registers and are used for memory protection and for base relative addressing. Memory protection is provided by checking the memory address in FA with the BR (Base Register) and the LR (Limit Register) for all memory cycles. Any address outside these bounds is flagged in the CD-register (CD(0) WRITE/SWAP, CD(1) READ). A memory read operation is always whether inside or outside the boundary, but WRITE or SWAP operatiors are allowed outside the boundary only if the override bit of the CD(2) register is true. A memory address equal to the BR or LR is consicered in bounds. Note: The memory protection is provided only on the initial pointer and does not provide protection on those memory bits accessed when the Field Length is greater than one. If the field is outside of the administrative memory, then PERM(2) (bit 2 of the PERM register) is set. The COUNT operation specified by the count variants will take

place regardless of whether or not the memory cycle takes place.

BR-register is compared with the FA-register in the cassette control micro to either cause a halt or a skip, depending upon the variant (see section 3.4.30).

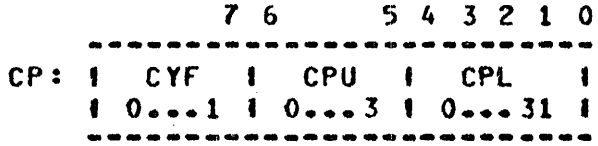
Each register is addressable as a source and as a sink.

3.1.13 C

The C (control) Register is a 24-bit register which is not addressable is an entity but which is functionally divided into one 8-bit section and four 4-bit sections.

3.1.13.1 CP (CYF, CPU, CPL)

The 8-bit section, addressed as CP, is comprised of the arithmetic unit carry flip-flop (CYF), the 2-bit unit control for the arithmetic unit (CPU) and the 5-bit variable data length control (CPL). CPU is not addressable. CP[6], the MSB of CPU, has no special meaning. CP[5], the LSB of CPU, determines the Unit of the ALU thus: 0: binary, 1: BCD. CP[7], CYF, ia also available as a source of referencing BICN[2].



3.1.13.2 CA, CB, CC, CD

The remaining 16-bits of the C-register are addressable in 4-bit groups as CA, CB, CC and CD. Their contents are available for aralysis and alteration via the 4-bit function box. MANIPULATE, SKIP and BIT-TEST-BRANCH instructions are applicable.

The 4-bit groups designated as CA and CB have no special functional assignment and are available as general purpose 4-bit storage registers.

The two 4-bit registers designatec as CC and CD are used fcr the storage of various processor states and conditions as shown belcw:

	3	2	1	0:LSB
CC:	CONTROL PANEL STATE LAMP FLIP-FLOP	100 MSEC REAL TIME CLOCK INTERRUPT	I/O BUS SERVICE REQUEST	CONTROL PANEL INTERRUPT

CC(3): The control panel state lamp flip-flop when true will cause the STATE lamp to light on the Diagnostic and Maintenance Console.

CC(2): The real time clock interrupt signal is developed from the primary power frequency, field adjustable for either 50Hz or 60Hz. The interrupt signal is received and is used to set the CC(2) bit once every 100 milliseconds.

CC(1): The I/O Bus service request interrupt level is derived from the various I/O controls connected to the processor's I/O Bus. The level is the result of a service request by one or more controls and is used to set the interrupt bit every clock time.

CC(0): The control panel interrupt level is derived from the on position of the control panel's INTERRUPT switch. The level from the switch is used to set the interrupt bit every clock time. This flip-flop also drives the lamp behind the INTERRUPT switch on the operators panel.

	3	2	1	0:LSB
CD:	Memory Read Data Error Interrupt	Memory Write/Swap Address (LR/BR Check) Out of Bounds Override	Memory Read Address (LR/BR Check) Out of Bounds Interrupt	Memory Write/Swap Address (LR/BR Check) Out of Bounds Interrupt

CD(3): There are several ways this bit can be manipulated. It is addressable as a 4-bit register and can be altered through the applicable micro-instructions. In addition, this flip-flop is recognized as the memory Read Data Error interrupt flag bit. As such, it can be set by:

M-register parity error,
Cache Key parity error,
Cache Double Hit,

Cassette parity error,
Uncorrectable S-memory (CPU) error,
Error log register has changed,
S-memory field out of bounds, and
S-memory micro has timed out.

CD(2), CD(1), CD(0):

The memory address out-of bounds signals are derivec from logic which compares the contents of the FA-register with the contents of the Base (BR) and Limit (LR) Registers cn all memory accesses. The state of the out-of-bounds override control bit does not affect the setting of out-cf-bcuncs interrupt bits but does affect the occurrence of the subsequent write operation. (See Section 3.1.12).

No reaction occurs as a result of any interrupt until the micro-program tests the interrupt bit.

A micro-instruction or the control panel CLEAR pushbutton is capable of resetting a bit in the C-register. The bit being reset will be false for at least one clock period following the reset regardless of the continued existerce of the condition to set the bit (e.g., control panel of serivce request interrupts). Any test micro-operator executec in this clock period will find the bit false. If the conditior does not continue to exist beyond the reset time, a failure to set the bit may occur (e.g., timer interrupt).

3.1.14 MAXS

MAXS is a 24-bit pseudo register that can be field-adjusted tc give the size (administrative or actual physical) of the S-memory installed in the system. It is addressable as a source only. MAXS, for main (S) memory, has 8K-byte resolution (least significant 16 bits are always zeroes).

3.1.15 U

The U-register is a 16-bit register used primarily to accumulate the bit-by-bit input from the control panel's tape cassette. The U-register is addressable as a source register only.

In RUN mode, if data is not yet available in the register, the micro-operator will be delayed. Data not accepted in time will be lost.

In TAPE mode, the register's contents are automatically moved to the M-register for execution except when executing micro-operators that reference the U-register as a source. In these cases the source data is moved directly from the U-register to the destination and will not be treated as an instruction to be executed. The next instruction from the tape will follow this data. In the tape mode, it is possible to move data to the M-register. This will result in a bit-OR of the moved data to M with the next micro being fetched to the M-register. For MOVE 24-BIT LITERAL (9C), the 8 bits of the literal in the M-register are also moved to the destination. For CONDITIONAL BRANCH, the A-register may be changed but the next micro-operator read from the tape will also be executed. Note: In RUN mode, the U-register may not be addressed after the issuance of a CASSETTE STOP micro.

3.1.16 DATA

DATA is a 24-bit pseudo register that can act as a source or as a destination. It is used to transfer data to and from the I/O Bus. When it is used as a source, the processor generates the RC (RESPONSE COMPLETE) signal to the interface and accepts the 24-bits of cata from the bus. When used as a destination, the processor generates the RC signal to the interface and the data from the designated source to the bus.

The micros which can source DATA are 1C and 2C; the micros which can sink to DATA are 1C, 2C, and 10C. The hardware prevents multiple RC's from occuring any sooner than 8 clocks apart.

Data can be executed from the console similar to other registers. However, because of the single RC that is generated, it is ciscussed separately. For review, in the console mode, the selected register is always sourced to the console lights. In order to cause the switches to be loaded into the register, the LOAD button must be activated. In the console mode, an RC is generatec only when the LOAD button is activated and inhibited at other times.

Assuming that a previous console command was an I/O command, then the I/O is either ready to accept or transmit data. If the I/C is to transmit data, then the selected I/O will hold the data on the MEX which is then displayed on the panel lights because nc RC is generated. Pushing the LOAD button will transmit an RC (processor switched commands to now transmit to the I/O) which will complete the operation and cause the I/O cata not be be displayed. If however, the I/O is ready to accept data (the processor must switch commands from sourcing to sinking into the I/O) then at the activation of the LCAD button, an RC would be generated and the switch contents will be strobed into the selected I/O device.

3.1.17 CMND

CMND (Command) is a 24-bit pseudo register that can act as a destination only. It is used to transfer commands to devices on the I/O Bus. The processor generates the CA signal to the interface and moves the data (or the command) from the designated source to the bus.

The micros which can sink CMND are 1C, 2C, and 10C. The hardware prevents a CA to follow an RC any sooner than 4 clocks.

CMND can also be executed from the console. Since CMND is a sink only, sourcing CMND will result in a no-operation. Then activating the LOAD button will cause a CA to be transmitted along with the transfer of the console switch contents to the I/O device

3.1.18 NULL

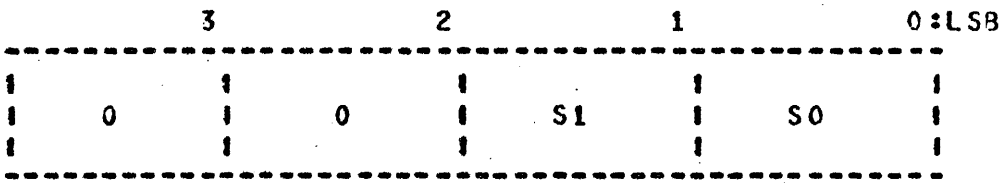
NULL is a 24-bit pseudo register that can act as a source and as a destination. When addressed as a source, all zeros are supplied to the destination. When addressed as a destination, the source data is not accepted. However, NULL is useful as a destination in order to pop the TAS register without affecting other registers.

3.1.19 CSW

CSW is a 24-bit pseudo register that can act as a source only. When addressed as a source, information on the positions of the control panel switches is supplied to the destination.

3.1.20 MSSW

The MSSW, micro-instruction source switch register, is a source/sink register. It is special in that the console micro-instruction source switch is bit-ORed onto the output of this register. Any value can be loaded into this register, but its contents are interpreted as the combination of the bits in the register and the console switch position.



During RUN, the source of the micro-instructions is controlled by MSSW. Specifically, this is where the instructions are sourced:

S1	S0	
00		The source of the micro-instructions is from the cache. However, if there is a MISS, the micro-instruction is directly from the S-memory as it routed to the cache. This mode is referred to as the normal mode where both the cache and S-memory are active.
01		The source of the micro-instructions is from the S-memory. The cache is disabled as a source of instructions.
10		The source of the micro-instructions is from cache. A MISS will result the processor halting. No instructions are available from the S-memory.
11		The M-register is frozen. The present contents of the M-register are repeated.

The MSSW can be sourced or sinked as a 4-bit register. All the move instructions which are applicable to a 4-bit register are also applicable to this register. The register has only two bits. Therefore, sourcing 4-bits will result in the upper 2-bits being a zero and sinking 4-bits will result in the upper 2-bits being lost.

Care must be used when altering this register since it affects the source of micro-instructions. Since the MP-3 pipeline is affected by the source of the micro-instruction, improper source switching may cause the pipeline to get out of synchronism.

3.1.21 NULLA, NULLB, NULLC, NULLD, and NULLE

These five pseudo registers can act as sources or as destinations. When addressed as a source, all zeroes are supplied to the destination. When addressed as a destination, the source data is not accepted.

NULLA replaces TOPM, NULLB replaces MSM, NULLC replaces MAXM, and NULLD replaces MBR of the Model I and II M-processors.

3.1.22 RESERVED

This pseudo register acts as a NULL register. It reserves the space the Liege LCR uses for IOR.

3.2 24-BIT FUNCTION BOX

The 24-bit "function box" is composed of a 24-bit arithmetic unit and a 24-bit combinatorial unit. It has as data inputs the contents of the X and Y-registers and the carry flip-flop (CYF). It also uses CPU (control for the arithmetic unit) and CPL (the 5-bit variable operand length) from the CP portion of the C-register.

All results from the combinatorial section are generated immediately and are continuously available to the micro-programmer. A move to one of the input registers or an alteration of a value in the CP portion of the C-register immediately generates a new result. The results are available to the next micro-instruction and are accessed by moving the contents of a result register to a destination register or by testing one of the 4-bit condition registers.

The results are most of the commonly used functions between two operands. These include the And, Or, Exclusive-Or, sum, carry-out, difference, and borrow functions, and the set of equal-to, greater-than, and less-than relationals. The results of the unary operations of complementation and masking are also available.

The results of the arithmetic unit are under control of the CPU and the CPL registers as follows:

CPU	UNIT TYPE	POSSIBLE CPL VALUES	DATA TYPE
00	1-bit operands	1 to 24	Binary
01	4-bit operands	4,8,12,16,20, or 24	4-bit binary (BCD)
10	1-bit operands	1 to 24	Binary
11	4-bit operands	4,8,12,16,20, or 24	4-bit binary (BCD)

For valid arithmetic operations, the operand length (as specified by CPL) must be an exact multiple of the length of the unit specified by CPU.

The contents of each of the registers described in the following subsections are immediately available to the micro-programmer.

3.2.1 SUM

SUM is a pseudo register equal to the sum of the X, Y, and CYF registers (X + Y + CYF). Zero bits in the more significant bit positions of the 24-bit result are produced when the length as determined by CPL (5-bit variable data length control) is less than 24. Results are not defined for CPL values 25 through 31. The carry-out level is generated from the bit position of the output specified by CPL. If CPL = 0, the carry-out level is equal to CYF. If CPL = 1, the carry-out level is generated from the rightmost bit of X, Y, and CYF. See Section 3.1.13.1, CP (CYF, CPU, CPL).

If CPU (2-bit arithmetic unit control) = 00, the binary sum is produced. See Section 3.1.13.1, CP (CYF, CPU, CPL).

If CPU = 01, the decimal sum is produced by considering the X and Y inputs to be comprised of six 4-bit units. Results are are not defined for non-Binary Coded Decimal (BCD) units. CPL must be a multiple of four.

If CPU = 10 or 11, the sum is defined as though it were 00 or 01 respectively. The ALU is controlled by CREG(05).

3.2.2 DIFF

DIFF is a pseudo register equal to the difference of the X, Y, and CYF registers (X - Y - CYF). Zero bits appear in the more significant bit positions of the 24-bit result when the length as determined by CPL is less than 24. Results are not defined for CPL values 25 through 31. The borrow-out level, generated from the static comparison of all 24-bits of X & Y, is true if X < Y or if X = Y and CYF is true.

If CPU = 00, the binary difference is produced.

If CPU = 01, the decimal difference is produced by considering the X & Y inputs to be comprised of six 4-bit units. Results are not defined for non-BCD units. CPL must be a multiple of four.

If CPU = 10 or 11, the difference is defined as though it were 00 or 01 respectively. The ALU is controlled by CREG(05).

A negative result is in 2's complement form in the binary case and in 10's complement form in the decimal case.

3.2.3 XANY, XORY, XEOY

XANY, (X and Y), XORY (X or Y), and XEOY (X exculsive-or Y) are 24-bit pseudo registers that hold the results of the specified logical functions. Zero bits are produced in the more significart bit positions of the 24-bit result when the length as determined by CPL is less than 24. Results for the CPL values 25 through 31 are not defined.

3.2.4 CMPX, CMPY

CMPX (complement of X) and CMPY (complement of Y) are 24-bit pseuco registers that hold the results of the specified logical functions. Zero bits are produced in the more significant positions of the 24-bit result when the length as determinec by CPL is less than 24. CPL values 25 through 31 have undefined results.

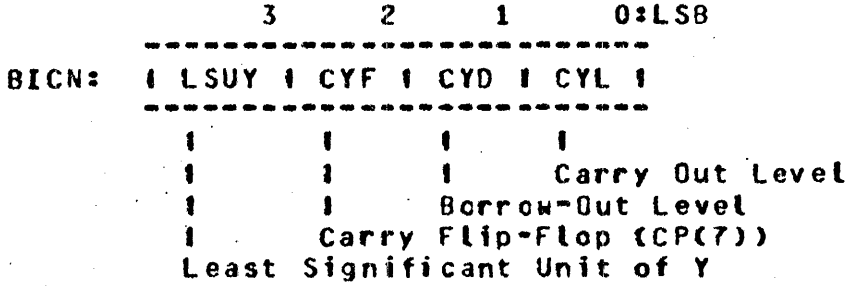
3.2.5 MSKX,MSKY

MSKX, (mask of X) and MSKY (mask of Y) are 24-bit pseudo registers that hold the mask of the appropriate register (X or Y). Beginning with LSB of X or Y, the number of bit positions included in the mask is determined by the value of CPL. Zero bits are produced in the more significant bit positions of the 24-bit result when the length as determined by CPL is less that 24. Results are not defined for CPL values 25 through 31.

3.2.6 BICN

BICN (binary conditions) is a 4-bit pseudo register that holds the following binary conditions, considered as a 4-bit group, and addressable as a source only.

NOTE: CYF is also addressed by the SET CYF M-instruction as well as being available in the (8-bit) group addressed as CP.



The carry-out level is a function of X, Y, CPL, and CPL. See Section 3.2.1. The borrow-out level is a function of X, Y, and CYF. See Section 3.2.2, DIFF.

LSUY is true if the least significant unit of Y is equal to 1 and CPU = 00 or 10, or if the least significant unit of Y is equal to 1001 and CPU = 01 or 11. Only CREG(05) of the CPU controls this operation.

3.2.7 XYCN

XYCN (XY condition) is a 4-bit pseudo register that holds the following relational conditions, considered as a 4-bit group and addressable as a source only:

	3	2	1	0:LSB
XYCN:	MSBX	X=Y	X<Y	X>Y

MSBX is true if the bit in X referenced by CPL is 1. CPL = 1 references the rightmost bit of X while CPL = 24 references the leftmost bit. MSBX = 0 if CPL = 0.

The relational results are based on the binary value of all 24-bits of X and Y.

3.2.8 XYST

XYST (XY states) is a 4-bit pseudo register that holds the following relational conditions, considered as a 4-bit group and addressable as a source only:

	3	2	1	0:LSB
XYST:	LSUX	INT	Y neq 0	X neq 0

LSUX is true when the least significant unit of X is equal to 1 and CPU = 00 or 10, or when the least significant unit of X is equal to 1001 and CPU = 01 or 11. Only CREG(05) bit of the CPU controls this operation.

The relational results are based on the binary value of all 24 bits of X or Y.

INT is true if any of the following conditions as reflected in INCN, CC, and CD are true. (See Section 3.2.9 INCN and 3.1.13.2 CA, CB, CC, CD.

INCN(3)	:	Missing Port Device
INCN(1)	:	Port Interrupt
CC(2)	:	100MS Real Time Clock Interrupt
CC(1)	:	Bus I/O Serivce Request Interrupt
CC(0)	:	Control Panel Interrupt
CD(3)	:	Memory Read Data Error Interrupt
CD(0)	:	Memory Write/Swap Address (LR/BR Check) Out of Bounds Interrupt

3.2.9 INCN

INCN (interrupt conditions) is a 4-bit pseudo register, source only, that reflects the state of certain interface lines between the processor and the port interchange (if present).

	3	2	1	0:LSB
INCN: (Port Connect)	PORT MISSING DEVICE	PORT HI PRIORITY INTERRUPT	PORT INTERRUPT	PORT LOCKOUT

When the port interchange is not present certain bits of INCN are strapped at a TRUE or FALSE value as shown below. See also the DISPATCH micro (3.4.29).

	3	2	1	0:LSB
INCN: (Direct Connect)	-----	FALSE	FALSE	FALSE

3.2.10 PERM (PARITY ERROR MEMORY)

This is a 4-bit register which indicates that a problem has occurred in memory. The register is defined as follows:

	3	2	1	0:LSB
PERM:	S-memory Micro-	S-memory field out-of-	S-memory Error Log	S-memory uncorrectable

Instruction Time-Out	bounds of the administrative memory	has changed	error during a Processor operation

This register gets set to zero whenever the machine is started from the halt state or in the halt state, whenever the register select is in column 6 and the LOAD button is activated.

PERM(3): All memory micros will be checked for time-out. These micros are Read/Write Memory (7C), Swap Memory (2D), Diagnostic Read/Write Memory (11D), and Dispatch (1E). The occurrance of a memory time-out will cause this bit to be set and immediately halt the processor with the stuck memory micro executing. This bit true indicates that an S-memory micro has timed-out.

PERM(2): This bit true indicates on any memory micros, either Read/Write, that the administrative memory (memory size as either determined by MAXS or the physical memory present) has been exceeded. For example, if the memory operation caused three stacks to be read, but the administrative memory had two stacks, then this bit would be set true. For the stack which was absent, all zeros would be returned, and there would be no error correcting action and hence no error indicators. During a fetch, a field out-of-bounds error will cause the machine to halt.

PERM(1): This bit will be set true whenever there has been a change to the error-log register. The error-log register is empty (reset, or cleared) after it has been read by the processor. The error-log register can be changed depending upon the level of the change information. Generally, the first error detected is logged. Then if there is an error of greater importance, its status is written into the error-log. There are three levels of errors, and these are the possible chain of events:

Case 1 single bit error which was corrected, replaced by an uncorrectable error from a non CPU device replaced by an uncorrectable error from the CPU access.

Case 2 an uncorrectable error from a non CPU device, replaced by a an uncorrectable error from the CPU access.

Case 3 an uncorrectable error from a CPU access. Note: The CPU error could be caused by a read error before a write operation.

Just for information, the error-log register contains the following information (see Section 3.4.35 for format):

1) Board number (4 bits) and chip row (2 bits).

2) Syndrome (6 bits) which indicates the type of error. See attached Table 3.

3) Write operation - the recorded error was the result of a write operation (read prior to the write; also may be either the bits which were written over or the bits which were restored).

4) Mutiple occurrence error. There was more than one occurrence of each type of error. An error status was lost.

5) Corrected single-bit error. One error which was recorded was a corrected single-bit error. Note: If no other higher bit error flag is set, then the error represented in the syndrome is a corrected single-bit error

6) Non CPU uncorrectable error. One error which was captured was a non CPU uncorrectable error. Again, if there is no CPU error recorded, then the syndrome contains the status for this error.

7) Uncorrectable CPU error. The error represented by the syndrome is the first CPU uncorrectable error.

PERM(0): This bit true indicates that there was an uncorrectable error as a result of a CPU access. During a fetch, an uncorrectable CPU error will result in a processor halt.

The PERM register can be a 4-bit source or sink for the move instruction. The conditions which set these bits also affect CD(3) and the RUN mode of the processor. This is illustrated by the diagram in Figure 28.

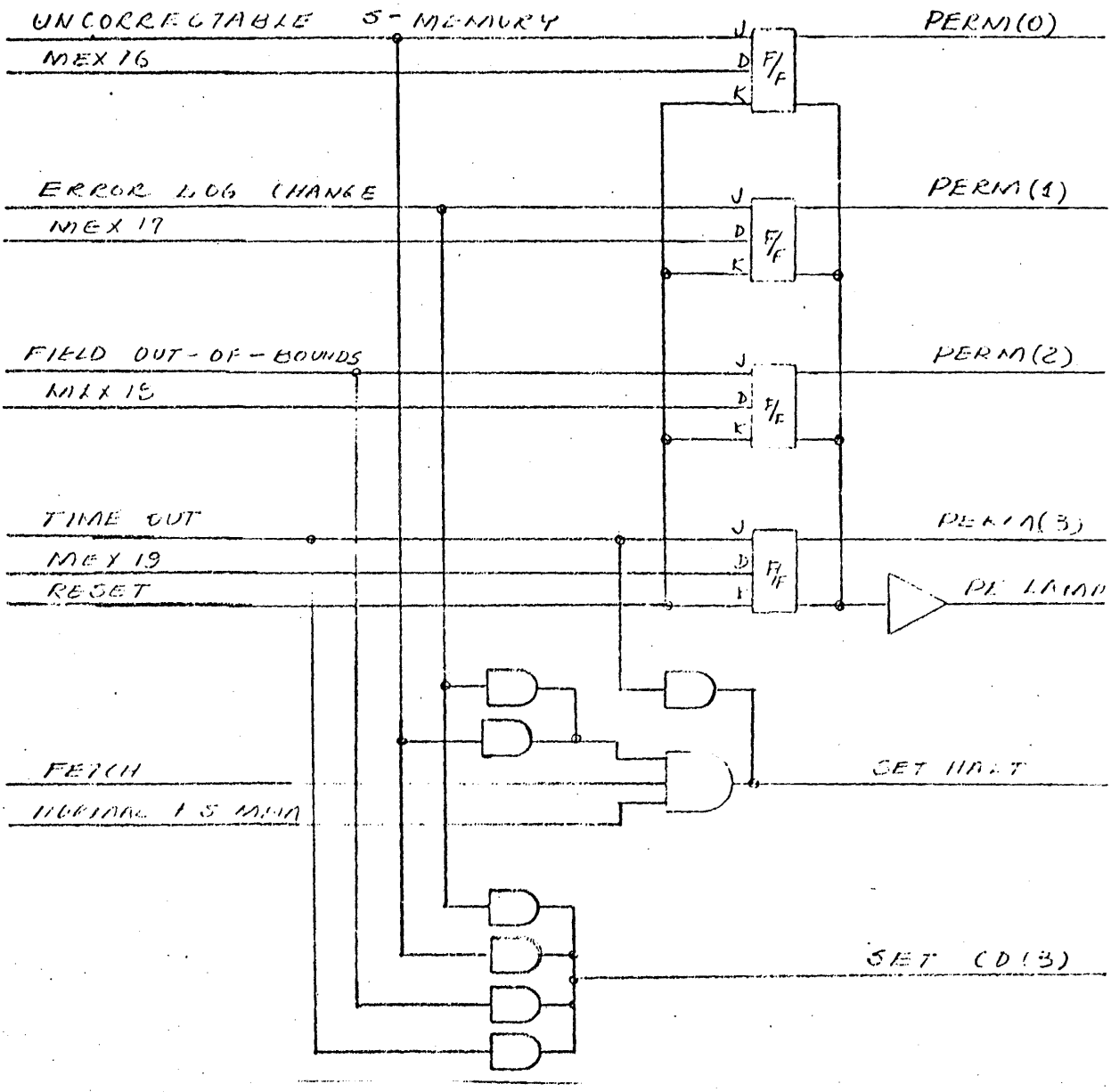

UNCORRECTABLE S-MEMORY
MEX 16
ERROR LOG CHANGE
MEX 17
FIELD OUT-OF-BOUNDS
MEX 15
TIME OUT
MEX 19
RESET
FETCH
J
D
K
F/F
PERM(0)
PERM(1)
PERM(2)
SET CD(3)

Fig 28

FIGURE 28

3.2.11 PERP (Parity Error Processor)

PERP is a 4-bit register which contains the error conditions which are generated in the processor. These errors are the result of the micro fetch mechanism, cassette tape read parity error, cache key parity error, parity error on the micro in the M-register, and a double hit in the cache read.

Specifically, the PERP register has this format:

	3	2	1	0:LSB
PERP:	Cache Double Hit	Cache Key Parity Error on Key A or Key B	Parity Error on the word fetched to the M-register	Cassette Read Error which canrot be corrected

From the HALT state, starting the machine will result in this register getting set to zero. Also, when halted and the register select is on column 6 and the LOAD button is activated, this register is set to zero.

PERP(3) Cache Double Hit

Cache double hit will result in both block A and block B read data from the cache to the merged. During fetch, this represents a hardware malfunction and the CPU will halt (cache enabled). Double hit can occur with the hardware in its normal operational state without any harware malfunctioning. This is when the key in both block A and B for a particular index is identical, and this occurs when writing into the cache, the A-register is the same for writing into block A and B. With the cache disabled, the double hit will not halt the processor but only set CD(3)and PERP(3).

PERP(2) Cache Key Parity Error on Key A or Key B

This bit is set whenever there is a parity error on reading the key store. Since both key stores, A and B, are read together, this flag can be set whenever either generates a bad parity check. The parity check is an odd parity over the key (8 bits), validity (1 bit), and parity (1 bit). When the MP-3 is cleared, the key is zero, valicity is one, and the parity is zero. With cache enabled, cache key parity error during a fetch will halt the MP-3; when disabled, it will set

CD(3) and PERP(2).

PERP(1) Parity Error on the M-register

This bit will indicate whenever there is a data parity error from the cache or a micro-fetch direcly from the S-memory. Since the MP-3 can move data to the M-register, at those times, the parity error reporting on the M-register will be disabled. When loacing the M-register from the console, anc when loading micros from the cassette, the parity bit is not transmitted. Therefore, durirg the tape mode and console loading, there will be no parity checking on the M-register. The conditions to cause this PERP(1) to be set will also halt the MP-3.

PERP(0) Cassette Read Error which cannot be corrected

This bit when true indicates a cassette read error. It will cause the MP-3 to halt whenever it is true during the TAPE (MTR) mode.

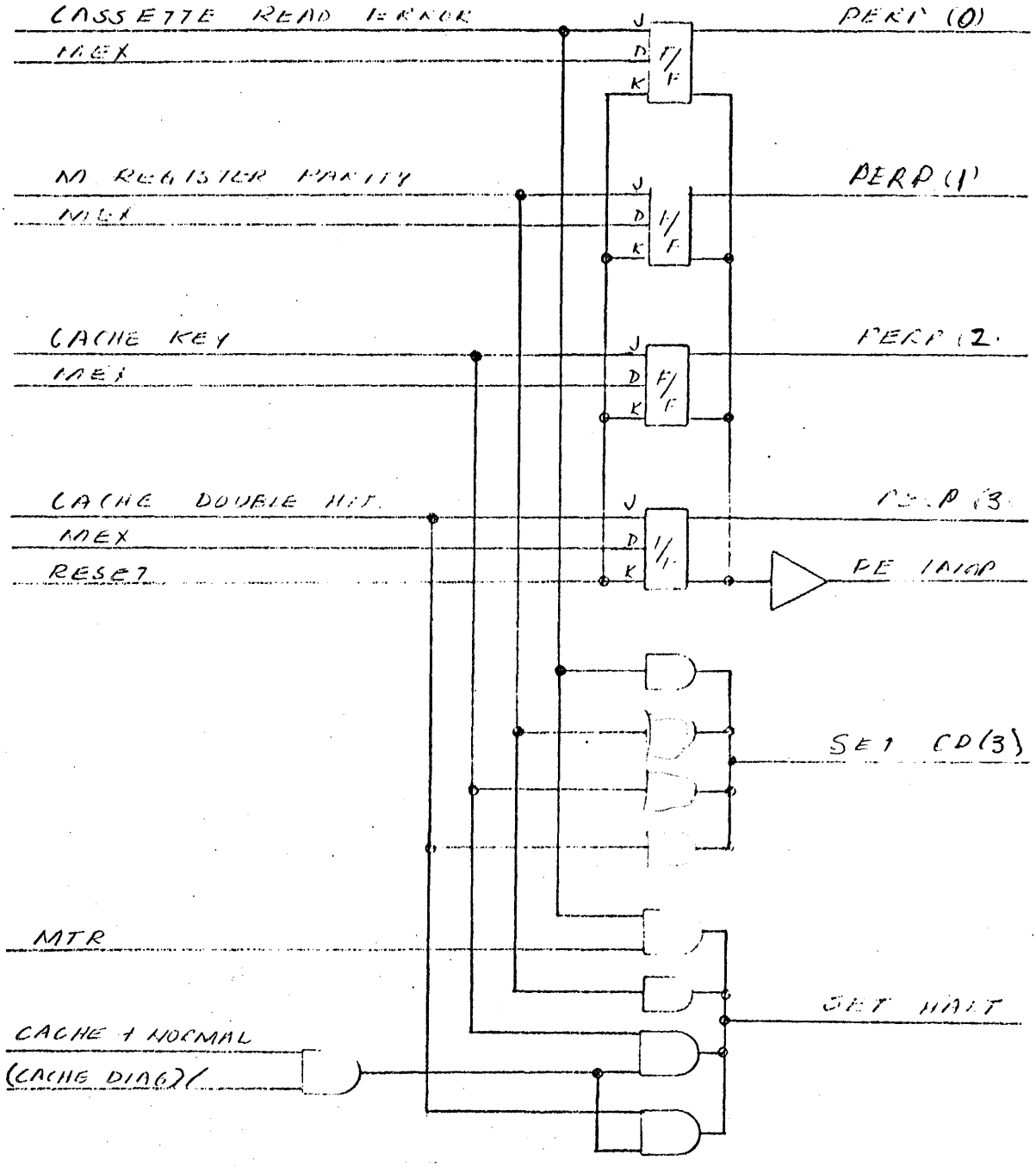
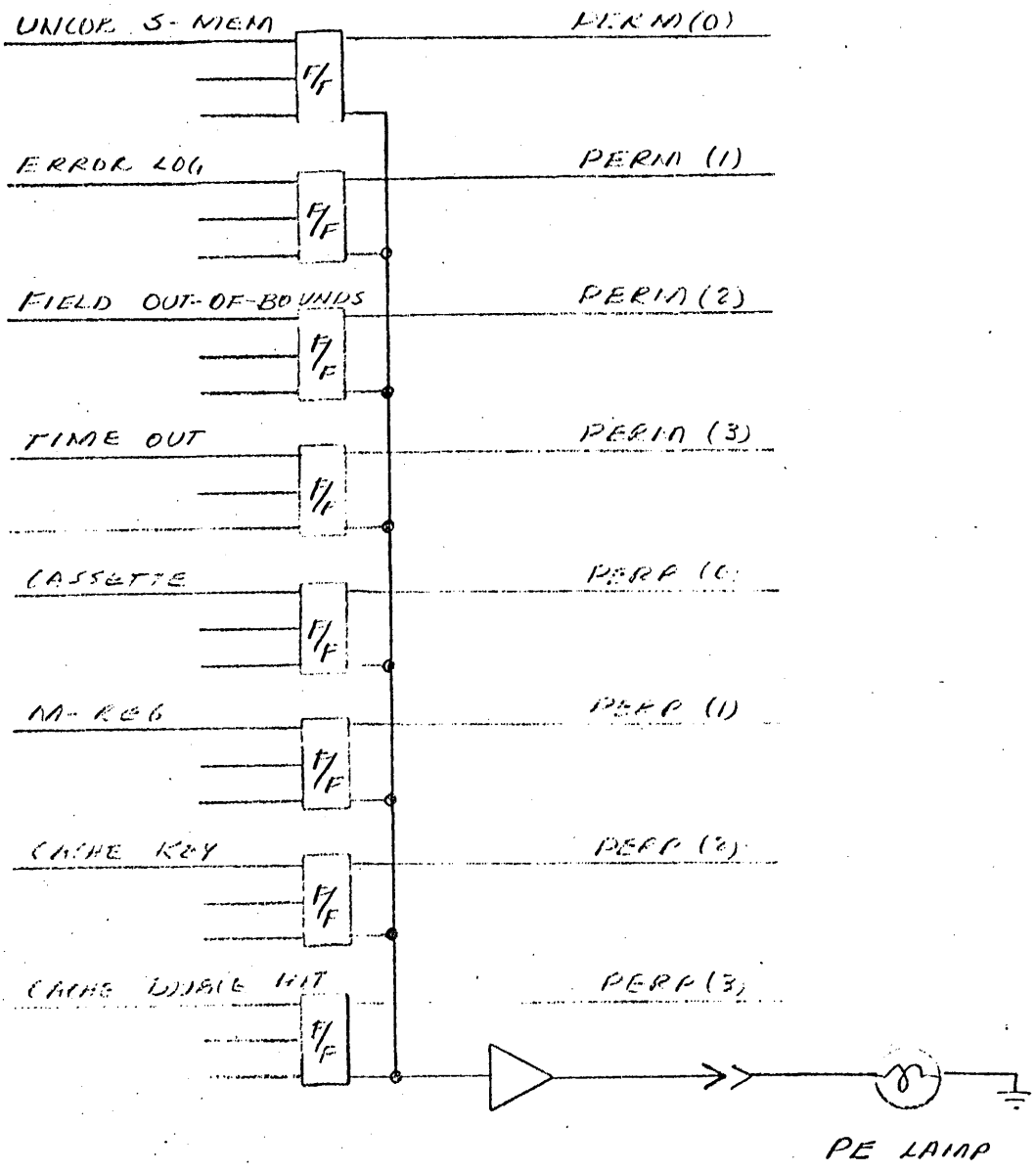

CASSETTE READ ERROR
MEX
M REGISTER PARITY
MEX
CACHE KEY
MEX
CACHE DOUBLE HIT
MEX
RESET
MTR
CACHE + NORMAL
(CACHE DIAG)
J
D
K
F/F
PERR (0)
PERR (1)
PERR (2)
PERR (3)
PE IAIAP
SET CD(3)
SET WAIT

FIGURE 29

The Parity Error Lamp on the console will be driven whenever there is any condition to set CD(3) except loading into CD(3). That is, the parity error lamp will be lit whenever any of the conditions to set the PERM or PERP register comes true.

The parity error lamp will go off whenever the machine is halted and the register select is in column 6 and the LOAD button is activated, or whenever the machine is started from the halt state.



3.2.12 TIME

TIME is a 24-bit register which continuously counts every 3 system clocks. It counts up and will wrap around. It is a source only register. However, attempting to move data into this register will cause it to be reset to zero. TIME can only be read by the Register Move (1C) micro; reset by Move 8-bit Literal (8C), Move 24-bit Literal (9C) and Shift/Rotate T-register (10C). No other micros will affect the TIME register.

3.3 4-BIT FUNCTION BOX

The 4-bit function box (4-bit arithmetic and combinatorial section of the processor) can accept, as one of its inputs, the conterts of any of the following 4-bit registers and pseudo registers. The second input is obtained from the M-instruction itself.

TA	TB	TC	TD	TE	TF	PERP
LA	LB	LC	LD	LE	LF	RESERVED
FU	FT	FLC	FLD	FLE	FLF	
CA	CB	CC	CD	NULLA	NULLE	
BICN	XYCN	XYST	FLCN	INCN	PERM	

Outputs include the result of most of the commonly used furctions between two operands; for example: set, and, or, exclusive-or, and binary sum and difference (both modulo 16). Outputs are are directec back to the source register if the source register is not a pseudo register.

The sum and difference output can be tested for overflcw and underflow respectively and, based on the test, a skip cf one instruction can be made.

The 4-bit function box also provides for the selective testing of one of the bits of a four-bit group and relative branching based on the result of the test. A skip of one instruction based on the result of testing on a combination of up to four bits in the group is also provided.

BICN, XYCN, XYST, FLCN, and INCN are not actually registers but can be sourced as if they were. They can be changed only as a result of changing the condition which they reflect.

3.4 M-INSTRUCTIONS

3.4.1 REGISTER MOVE

FORMAT:	OP CODE 0001	SOURCE REGISTER GROUP # 0 -15	SOURCE REGISTER GROUP # 0...3	DESTINATION REGISTER SELECT # 0...3	DESTINATION REGISTER GROUP # 0...15

Move the contents of the source register to the destination register. If the source register is smaller than the destination register, data are right justified with left (most significant) zero bits supplied. If the source register is larger than the destination register, data are truncated from the left.

The contents of the source register are unchanged unless it is also the destination register.

Exceptions:

1) When M is used as a destination register in RUN, STEP, or TAPE mode, the operation is changed to a bit-OR which modifies the next micro-operation. It does not modify the instruction as stored in either the cache or the S-memory.

2) CMND is excluded as source register.

3) BICN, FLCN, XYCN, XYST, INCN, CSW, SUM, CMPX, CMPY, XANY, XEQY, MSKX, MSKY, XORY, DIFF, MAXS, and U are excluded as destination registers.

4) When DATA is designated as a source, CMND and DATA are excluded as destinations.

5) U is excluded as a source in STEP mode.

6) TIME is excluded as a sink register.

3.4.2 SCRATCHPAD MOVE

FORMAT:

OP CODE	RGSTR GRP #	RGSTR SLCT#	DIRECTION	SCRTCHPD WORD	SCRTCHPD WORD ADDRESS
0010	0...15	0...3	0 TO SCRTCHPD 1 FROM SCRTCHPD	0-LFT WRD 1-RT WRD	0...15

Move the contents of the register (SCRATCHPAD) to SCRATCHPAD (register). If the move is between registers of unequal lengths, the data is right justified with left (most significant) zero bits supplied or with data truncated from the left, whichever is appropriate.

The contents of the source register are unchanged.

Exceptions:

1) When M is used as a destination register, the operation is changed to a bit-or which modifies the next micro-operation. It does not modify the instruction as stored in either the cache or the S-memory.

2) CMND is excluded as source register.

3) BICN, FLCN, XYCN, XYST, INCN, CSW, SUM, CMPX, CMPY, XANY, XORY, XEOY, MSKX, MSKY, DIFF, MAXS, and U are excluded as destination registers.

4) U is excluded as a source in STEP mode.

5) TIME is excluded as SOURCE or SINK.

3.4.3 SWAP F WITH DOUBLEPAD WORD

FORMAT:

OP CODE 0000 0111	DESTINATION 48-BIT SCRATCHPAD WORD 0...15	SOURCE 48-BIT SCRATCHPAD WORD 0...15

Move the contents of the FA and FB-registers to a holding register. Move the contents of the left and right word of the source scratchpad word to the FA and FB-register respectively. Move the contents of the holding register to the left and right word of the destination scratchpad word.

3.4.4 STORE F INTO DOUBLEPAD WORD

FORMAT:

OP CODE 0000 0000 0100	SCRATCHPAD WORD ADDRESS 0...15

Move the contents of the FA and FB-register to the left and right word respectively of the designated scratchpad word.

The contents of the source registers are unchanged.

3.4.5 LOAD F FROM DOUBLEPAD WORD

FORMAT:

OP CODE 0000 0000 0101	SCRATCHPAD WORD ADDRESS 0...15

Move the contents of the left and right word of the designated scratchpad word to the FA and FB-register respectively.

The contents of the source registers are unchanged.

3.4.6 MOVE 8-BIT LITERAL

FORMAT:

OP CODE	DESTINATION REGISTER GROUP #	LITERAL
1000	0...15	0...255

Move the 8-bit literal given in the instruction to the destination register. If the move is between registers of unequal lengths, the data is right justified with left (most significant) zero bits supplied. The register select number is assumed to be 2.

Move to TIME-register will result in the TIME to reset to zero.

Exceptions:

1) CSW is excluded as destination register.

2) When M is used as a destination register, the operation is changed to a bit-or which modifies the next micro instruction. It does not modify the instruction as stored in either the cache or the S-memory.

3.4.7 MOVE 24-BIT LITERAL

FORMAT:

OP CODE	DESTINATION REGISTER GROUP #	24-BIT LITERAL
1001	0...15	0...MAX

Move the 24-bit literal given in the instruction to the destination register. If the move is between registers of unequal lengths, the literal is truncated from the left. The register select numter is assumed to be 2.

Move to TIME-register will result in the TIME to reset to zero.

Exceptions:

1) CSW and M are excluded as destination registers.

3.4.8 SWAP MEMORY

FORMAT:

OP CODE	REGISTER #	FIELD	MEMORY
0000 0010	00 = X 01 = Y 10 = T 11 = L	DIRECTION 0 - POSITIVE 1 - NEGATIVE	FIELD LENGTH 0...24

Swap data from main memory with the data in the specified register. If the value of the memory field is less than 24, the data from memory is right justified with left (most significant) zero bits supplied while the data from the register is truncated from the left.

Register FA contains the bit address of the memory field while the field direction sign and field length is given in the instruction.

If the value of the memory field length as given in the instruction is zero, the value given in CPL is used.

3.4.9 READ/WRITE MEMORY

FORMAT:

OP CODE	DIRECTION	COUNT	RGSTR #	FIELD	MEMORY
0111	0 to RGSTR 1 TO MEMORY	VRNTS 0...7	00 = X 01 = Y 10 = T 11 = L	DIRECTION 0 - POSITIVE 1 - NEGATIVE	FIELD LENGTH 0...26

Move the register's (memory's) contents to the memory (register). If the value of the memory field length is less than 24, the data from memory is right justified with left (most significant) zero bits supplied while the data from the register is truncated from the left.

The contents of the source is unchanged.

Register FA contains the bit address of the memory field while the memory field direction sign and memory field length is given in the instruction.

If the value of the memory field length as given in the instruction is zero, the value in CPL is used.

Memory field length values (or CPL values if MFL = 0) of 25 and 26 are truncated to the value 24. When used on a WRITE operation, the value 25 causes all error logging and reporting to be suppressed. Correct, from an ECC point of view, data is written into memory. When used on a WRITE operation, the value 26 causes the same action as the value 24.

For a description of the count variants, see Section 3.4.10, COUNT FA/FL.

3.4.10 COUNT FA/FL

FORMAT:

OP CODE	COUNT VARIANTS	LITERAL 0...31
0000 0110	0...7	

Increment (decrement) binarily the designated register(s) by the value of the literal contained in the instruction or by the value of CPL if the value of the literal is zero.

Neither overflow nor underflow of FA is detected. The value of FA may go through its maximum value or its minimum value and wrap around.

Overflow of FL is not detected. The value of FL will go through its maximum value and wrap around. Underflow of FL is detected and will not wrap around. The value of zero is left in FL.

Literal values (or CPL values if LIT = 0) of 25 through 31 are truncated to the value 24.

Count variants are as follows:

V = 000 NO COUNT
    001 COUNT FA UP
    010 COUNT FL UP
    011 COUNT FA UP AND FL DOWN
    100 COUNT FA DOWN AND FL UP
    101 COUNT FA DOWN
    110 COUNT FL DOWN
    111 COUNT FA DOWN AND FL DOWN

3.4.11 SCRATCHPAD RELATE FA

FORMAT:

OP CODE	RESERVED	SIGN OF SPAD WORD	LEFT SCRATCHPAD WORD ADDRESS
0000 1000	000	0-POSITIVE 1-NEGATIVE	0...15

Replace the contents of the FA-register by the binary sum of the FA-register and specified scratchpad register.

Neither overflow nor underflow of FA is detected. The value of FA may go through its maximum value or its minimum value and wrap around.

3.4.12 EXTRACT FROM REGISTER T

FORMAT:

OP CODE	ROTATE BIT COUNT	DESTINATION REGISTER	EXTRACT BIT COUNT
1011	0...24	00 - X 01 - Y 10 - T 11 - L	0...24

Rotate register T left by the number of bits specified and then extract from the right the number of bits specified. Move this result to the destination register supplying left most (most significant) zero bits if the extract count is less than 24.

The contents of the source register are unchanged unless it is also the destination register.

A rotate value of 24 is equivalent to 0.

3.4.13 SHIFT/ROTATE REGISTER T LEFT

FORMAT:	OP CODE 1010	DSTNATN REGISTR GROUP # 0...15	DSTNATN REGISTR SELECT# 0...3	S/R VARIANT 0 - SHIFT 1 - ROTATE	S/R BIT COUNT 0...24

SHIFT (ROTATE) Register I left by the number of bits specified and then move the 24-bit result to the destination register. If the move is between registers of equal lengths, the data is right justified with data truncated from the left.

The contents of the source register are unchanged unless the source register is also the destination register.

Zero fill on the right and truncation on the left occurs for the SHIFT operation.

If the value of the SHIFT/ROTATE count as given in the instruction is zero, the value given in CPL is used.

Exceptions:

1) When M is used as a destination register, the operation is changed to a bit-or which modifies the next micro-operation. It does not modify the instruction as stored in the memory.

2) BICN, FLCN, XYCN, XYST, INCN, CSW, SUM, CMPX, CMPY, XANY, XEQY, MSKX, MSKY, XORY, DIFF, MAXS, and U are excluded as destination registers.

3) TIME as a sink will reset TIME to zero

3.4.14 SHIFT/ROTATE REGISTER X/Y LEFT/RIGHT

FORMAT:	OP CODE 0000 0100	S/R VARIANT 0-SHIFT 1-ROTATE	L/R VARIANT 0-LEFT 1-RIGHT	X/Y VARIANT 0-X REG 1-Y REG	S/R BIT COUNT 0...24

SHIFT (ROTATE) Register X or Register Y left (right) by the number of bits specified.

Zero fill on the right and truncation on the left occurs for the left shift. Zero fill on the left and truncation on the right occurs for the right shift.

3.4.15 SHIFT/ROTATE REGISTERS XY LEFT/RIGHT

FORMAT:

OP CODE	S/R VARIANT	L/R VARIANT	S/R BIT COUNT
0000 0101	0-SHIFT 1-ROTATE	0-LEFT 1-RIGHT	0...48

SHIFT (ROTATE) Register X and Y left (right) by the number of bits specified. The register X is the leftmost (more significant) half of the concatenated 48-bit XY-register.

Zero fill on the right and truncation on the left occurs for the left shift. Zero fill on the left and truncation on the right occurs for the right shift.

3.4.16 NORMALIZE X

FORMAT:

OP CODE
0000 0000 0000 0011

SHIFT the X-register left while counting FL down, until FL=0 or until the bit in X referenced by CPL = 1. Zeros are shifted into the rightmost end of X.

CPL = 1 references the rightmost bit of X while CPL = 24 references the leftmost bit of X. CPL = 0 is undefined.

3.4.17 READ/WRITE CACHE

OP CODE	FORCE PARITY	VARIANTS

```
FORMAT: |                  | INTO KEY STORE |          |
        | 0000 0000 0111 |   0 = GOOD     |          |
        |                  |   1 = BAD      | 0......7 |
        ---------------------------------------------
```

The following variant codes cannot be implemented by the programmer:

```
000   CONSOLE  WRITE A
001   CONSOLE  WRITE B
010   CONSOLE READ MICROS
011   CONSOLE  READ KEYS
100   NOT ASSIGNED
```

The programmer should only execute these variant codes:

```
101   DIAGNOSTIC  WRITE
110   DIAGNOSTIC  READ  MICRO
111   DIAGNOSTIC  READ  KEYS
```

VARIANTS:

000 WRITE FROM CONSOLE ONE WORD INTO BLOCK A.

Write one word, 17 bits, from the console switches (CSW) into the cache. The 17 bits are defined as follows:

```
                  16   15
CSW:       ----------------------------------------
POSITIONS  |   PARITY   |   16 BIT WORD TO CACHE   |
           |   BIT*     |                          |
           ----------------------------------------
```

* ODD OVER 17 BITS

The A-register is used as the address into the cache. The A-register is interpreted as follows:

```
A-REG     21    14 13        6 5        4
          ---------------------------------
BITS      |  KEY   |   INDEX   |   WORD   |
          ---------------------------------
```

The Key Store A is written with KEY (A) and the VALIDITY (A) indicates a presence of a valid micro (validity bit = 0 indicates presence, validity bit = 1 indicates not present). This micro overrides the LRU, writing into Block A regardless of LRU. LRU is not changed as a result of this write operation.

The A-register is not incremented automatically by this operation. However, hitting the console "INC A" button will cause the A-register to be incremented by 1 in bit position 04.

VARIANTS:
---

001 WRITE FROM CONSOLE ONE WORD INTO BLOCK B.

This is the same as variant 000 except the word is written into Block B instead of Block A.

010 READ ONE WORD FROM THE CACHE TO THE CONSOLE LIGHTS.

The word in cache is addressed by the A-register. It is that word which is located at the at the associative match of the key as pointed to by the index. The validity bit for that key must also be false. The selection of the one word in the four-word block is determined by bits 04 and 05 of the A-register.

There must be a comparison or HIT of the key (8 bits) and the validity bit (1 bit). If there is no associative match (MISS), then all zeros will be sourced from the cache.

The LRU bit is set to the state representing the block which was not selected.

The data word in cache is 17 bits, 16 data and 1 parity. There is no parity check on the data read from cache since the data by-passes the M-register. However, there is a parity check on both keys A and B.

The A-register is not incremented automatically by the operation. However, hitting the console "INC A" button will cause the A-register to be incremented by 1 in bit position 04.

011 READ CACHE KEYS TO THE CONSOLE LIGHTS.

The index portion only (Bits 6-13) of the A-register is used as the address to read the keys. This is a non-associative read. The console lights will contain the following data:

23	22	21	20	19	18 11	10	9	8	1	0
LRU	D	HIT BLOCK B	HIT BLOCK A	B PARITY	KEY B	VALIDITY B	A PARITY	KEY A		VALIDITY A

Nothing is changed by this operation. LRU, A-register, Validity and Parity bits all remain unchanged.

There will be parity checks on both key fields for parity errors. If an error is found, PERP(2) will be set along with CD(3) ard the console parity error lamp.

101 DIAGNOSTIC CACHE WRITE

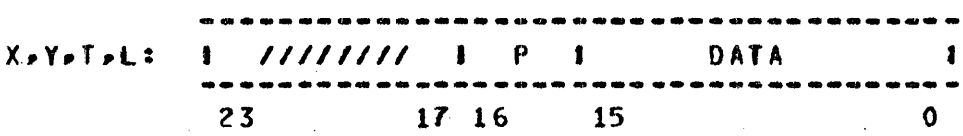
The FA-register contains the address to write into cache as cefined below:

The low order 6 bits and high order 2 bits are ignorec. LRU determines which block to write into. LRU does not change as a result of this operation; it can also force either good/bad parity into the cache key.

Four 17 bit words are written into cache. They are:

X - TO WORD 0
Y - TO WORD 1
T - TO WORD 2
L - TO WORD 3

The 17 bits are 16 data bits and 1 odd parity bit on 17 bits. The format is:



The sequence of operation is:

1) Save A,

2) FA-register to A-register,

3) Write X-register to cache, word D; Write Key(A) to cache (Key); Set valid bit in key; Generate and write parity in key,

4) Write Y-register to cache, word 1;

5) Write T-register to cache, word 2;

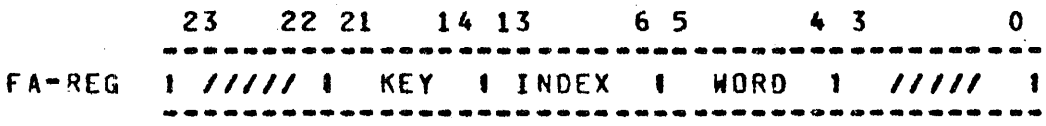
6) Write L-register to cache, word 3.

7) Restore A-register.

NOTE: The key information is the same for all write accesses.

110 DIAGNOSTIC CACHE READ DATA

Read one word (16 data bits, 1 parity bit) from cache to the X-register.



The FA-register contains the address of the word in cache. This is an associative read where the read occurs in the block where the valid bit is false and the keys match.

The LRU is set to the block which was not selected if there is a HIT.

The data word in cache is 17 bits, 16 data and 1 parity.

The sequence of operation is:

1) Save A-register,

2) Move FA-register to A-register,

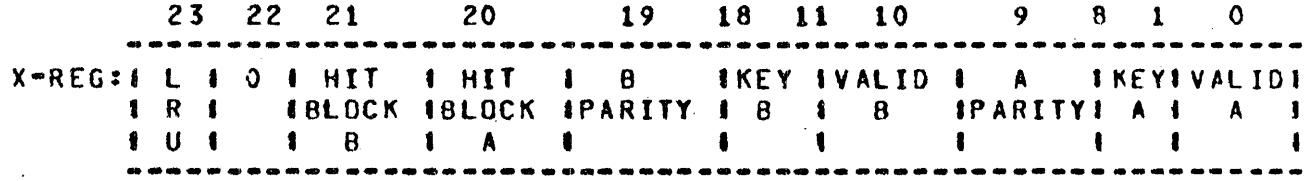
3) Read cache to X-register (17 bits),

4) Restore A-register.

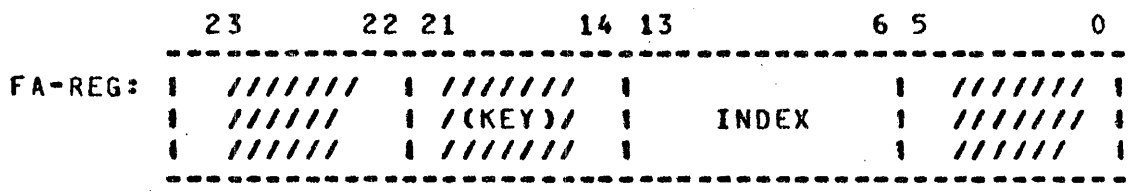
If there is a MISS, then all zeros will be read from the cache. There is no data parity check on this Read since the data by-passes the M-register. However, there is parity check on both keys A ard B.

111 DIAGNOSTIC CACHE READ KEYS

Both keys, valid bit, parity, and hit status in addition to the LRU is read to the X-register. The format of the data in the X-register is:



The FA-register is used to determine which keys to reac. The FA-register format is the same, but only the index portion is meaningful.



This is a non-associative read. Nothing in the cache is changed as a result of this operation (including the LRU).

The sequence of operation is:

1) Save A,

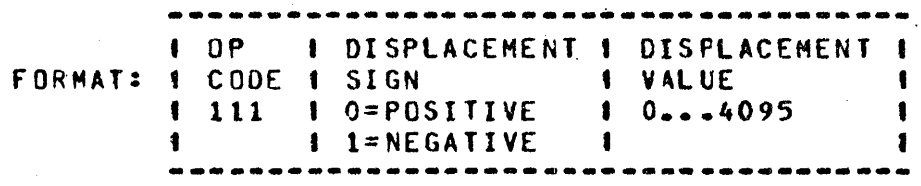
2) Move FA-register to A-register,

3) Move Keys (etc.) to X-register,

4) Restore A.

If there is key parity error, PERP(2) and CD(3) will be set.

3.4.18 CALL



Push the address of the next in-line micro-instruction into the A Stack and then fetch the next micro-instruction from the location obtained by adding the signed displacement value given in the instruction to the address of the next in-line micro-instructior.

A displacement value indicates the number of 16-bit words.

Note: When the A Address is stored in the A Stack, it is multiplied by 16 and stored as a bit address.

Note: Exit is accomplished by employing the REGISTER MOVE instruction with the TAS as the source register and A as the destination register.

3.4.19 BRANCH

FORMAT:

OP CODE	DISPLACEMENT SIGN	DISPLACEMENT VALUE
110	0=POSITIVE 1=NEGATIVE	0...4095

Fetch the next micro-instruction from the location obtaired by adding the signed displacement value given in the instruction to the address of the micro-instruction next-in-line.

A displacement value indicates the number of 16-bit words.

3.4.20 BIAS

FORMAT:

OP CODE	VARIANTS	TEST CPL NEQ 0 FLAG
0000 0000 0011	0...7	0 - NO TEST 1 - TEST CPL RESULT

Set CPU to the value 1 (01) if the value of FU is 4 or 8 and to 0 (00) otherwise, unless V = 2. If V = 2, the CPU value is determined by SFU in lieu of FU.

Set the value of CPL to the smallest of the values denoted or each line in the following table.

V	VALUES
0	FU
1	24 and FL
2	24 and SFL
3	24 and FL and SFL
4	CPL
5	24 and CPL and FL
6	CPL
7	CPL

If test flag equals 1 and final value of CPL is not zero, the next 16-bit micro-instruction is skipped.

3.4.21 SET CYF

FORMAT:

OP CODE	VARIANTS
0000 0000 0110	1,2,4,8

Set the carry flip-flop as specified by the variants.

V = 1 Set CYF to 0
2 Set CYF to 1
4 Set CYF to CYL
8 Set CYF to CYD

Note CYD = (X<Y) + (X=Y)CYF.

3.4.22 4-BIT MANIPULATE

FORMAT:

OP CODE	REGISTER GROUP #	REGISTER SELECT #	VARIANTS	LITERAL
0011	0...15	0...1	0...7	0...15

Perform the operation specified by the variants on the designated register.

V = 0 Set the register to the value of the literal.

1 Set the register to the logical And of the register and literal.

2 Set the register to the logical Or of the register and literal.

3 Set the register to the logical Exclusive-Or of the register and literal.

4 Set the register to the binary sum (modulo 16) of the register and literal.

5 Set the register to the binary sum (modulo 16) of the register and literal, and skip the next M-Instruction if a carry is produced.

6 Set the register to the binary difference (modulo 16) of the register and literal.

7 Set the register to the binary difference (modulo 16) of the register and literal, and skip the next M-Instruction if a borrow is produced.

Exception

BICN, FLCN, XYCN, XYST, and INCN, when specified as operand registers, are not changed as a result of this operation. However, the carry and borrow outputs are produced and a skip can result.

3.4.23 BIT TEST BRANCH FALSE

	OP CODE	REG GROUP #	REGISTER SELECT #	REG BIT #	DSPLCMNT SIGN	DSPLCMNT VALUE
FORMAT:	0100	0...15	0...1	0...3	0-POS 1-NEG	0...15

Test the designated bit within the specified register and branch relative to the next instruction by the signed displacement value if the bit is zero. If the bit is one, a displacement value of zero is assumed and control passes to the next in-line M-instruction. A displacement value indicates the number of 16-bit words from the next in-line instruction.

See Section 3.1.13 for information on the reset of bits in the C-register.

3.4.24 BIT TEST BRANCH TRUE

	OP CODE	RGSTR GROUP #	RGSTR SELECT #	RGSTR BIT #	DSPLCMNT SIGN	DSPLCMNT VALUE
FORMAT:	0101	0...15	0...1	0...3	0-POSITIVE 1-NEGATIVE	0...15

Test the designated bit within the specified register and branch relative to the next instruction by the signed displacement value if the bit is one. If the bit is zero, a displacement value of zero is assumed and control passes to the next in-line M-instruction. A displacement value indicates the number of 16-bit words from the next in-line instruction.

See Section 3.1.13.2, CA, CB, CC, CD, for information on the reset of bits in the C-register.

3.4.25 SKIP WHEN

FORMAT:

OP CODE	REGISTER GROUP #	REGISTER SELECT #	VARIANTS	MASK
0110	0...15	0...1	0...7	0...15

Test only the bits in the register that are referenced by the "1" bits in the mask, ignoring all others unless V = 2 or V = 6. If so, compare all bits for an equal condition. Then perform the action as specified below.

V = 0 If any of the referenced bits is a "1", skip the next M-instruction.

1 If all of the referenced bits are "1", skip the next M-instruction.

2 If the register is equal to the mask, skip the next M-instruction. SKIP GLOSS 3=Same as V = 1, but also clear the referenced bits to zero without affecting the non-referenced bits.

4 If any of the referenced bits is a "1", do not skip the next M-instruction.

5 If all of the referenced bits are "1" do not skip the next M-instruction.

6 If the register is equal to the mask, do not skip the next instruction.

7 Same as V = 4, but also clear the referenced bits to zero without affecting the non-referenced bits.

Note: If the mask equals 0000, the "ANY" result is false. The skip is not made for V = 0 and is made for V = 4. If the mask equals 0000, the "ALL" result is true. The skip is made for V = 1 and V = 3 and is not made for V = 5 and V = 7.

Exceptions:

1) BICN, FLCN, XYCN, XYST, and INCN cannot be cleared with V = 3 or 7. However, they can be tested.

2) See Section 3.1.13 related to the reset of bits in the C-register.

3.4.26 CLEAR REGISTERS

FORMAT:

OP CODE	REGISTER FLAGS 8 BITS							
0000 0011	L	T	Y	X	FALSE	FAIL	FUP	C P

Clear the specified register(s) to zero if the respective flag bit is a one.

3.4.27 BIND

FORMAT:

OP CODE
0000 0000 0000 0100

Move the 24-bit sum of the L and T-registers to the A-register. Since the A-register is 18 bits, the lower 4 and upper 2 bits of the 24-bit sum are lost.

3.4.28 OVERLAY M-MEMORY

FORMAT:

OP CODE
0000 0000 0000 0010

Overlay M-Memory from main memory. In this machine, this operator will be interpreted as a No-operation micro (3.4.32).

3.4.29 DISPATCH

FORMAT:

OP CODE	VARIANTS	SKIP VARIANT
0000 0000 0001	000-LOCKOUT	0-SKIP IF ALREADY LOCKED
	001-WRITE LOW	
	010-READ	1-SKIP IF NOT ALREADY LOCKED
	011-READ & CLEAR	
	100-WRITE HIGH	(Applies only to lockout variant)
	101-PORT ABSENT	

Dispatch operations are used to send/receive interrupt and interrupt information to/from other ports.

Since the interrupt system is shared by all ports, the processor should gain control of the interrupt system by successfully completing a LOCKOUT prior to a DISPATCH WRITE.

LOCKOUT sets the lockout bit in the DISPATCH Register and allows, via the skip variant, skipping or not skipping the next 16-bit instruction based upon success or failure (already set) of the LOCKOUT.

WRITE (High or Low) DISPATCH sets the Lockout and Interrupt flip flops in the port interchange. It also stores the contents of the L-register into memory 0 through 23 and the contents of the least significant 7 bits of the T-register (designating the destination port # and channel #) into the appropriate port interchange register. In addition, it sets (Write High) or resets (Write Low) the high Interrupt flip flop in the port interchange.

READ DISPATCH stores the contents of memory locations 0 through 23 into the L-register and the contents of the Port Channel register into the least significant 7 bits of the T-register. The other 17 bits of the T-register are unaffected.

READ AND CLEAR DISPATCH in addition to performing the READ DISPATCH operation clears the lockout flip flop, the two interrupt flip flops and the Port Device Absent flip flop in the port interchange. It does not clear any memory locations.

PORT ABSENT is executed by the processor when necessary to return a Port Device Absent level signal to another port indicating the absence of the designated channel.

Dispatch operations in the case of direct connect to memory are limited to the following:

1) LOCKOUT: Always skips.

2) WRITE LOW: Always sets Port Device Absent level true (true indicates absence).

3) READ & CLEAR: Always sets the Port Device Absent Level false (false indicates present).

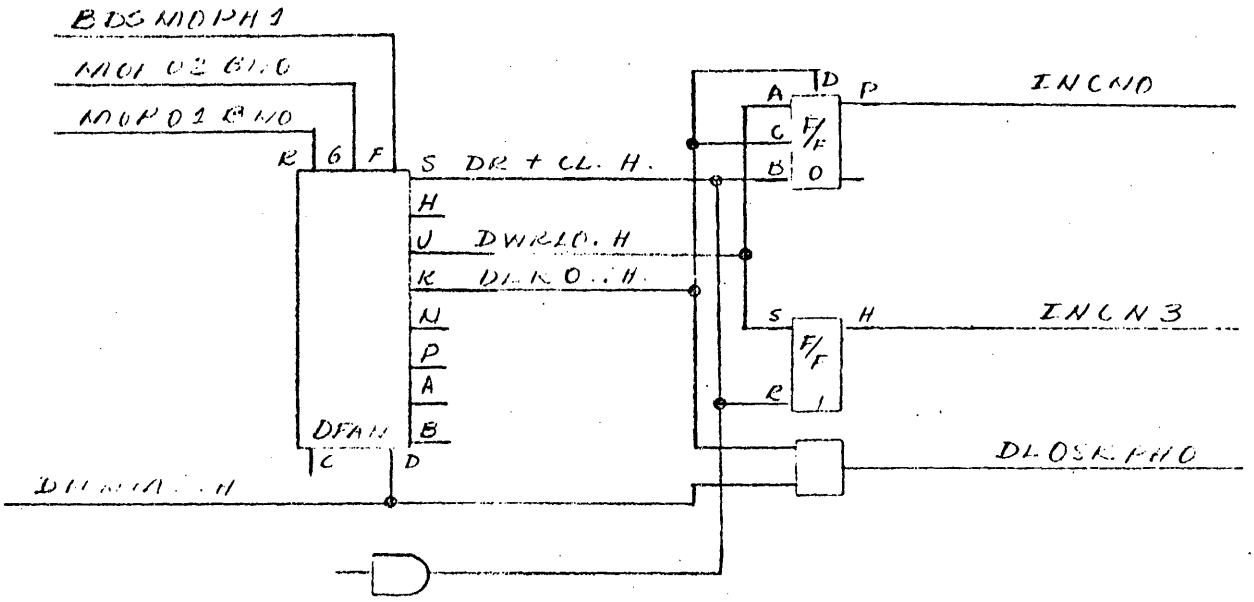


FIGURE 31 DISPATCH

BDSMOPH1 : DISPMATCH MICRO (M04-M15)

MOP02NBO : M02

MOP01BNO : M01

DNNM..H. : DONE MEMORY

INCNO : PORT LOCKOUT BIT

INCN3 : PORT MISSING DEVICE BIT

DR+CL.H. : DISPATCH READ AND CLEAR

DWRLO.H. : DISPATCH WRITE LOW

DLKO..H. : DISPATCH LOCKOUT SKIP

No changes occur in the T and L-registers. In the INCN-register only the Port Device Absent bit can change. The Lockout, the Interrupt, and High Priority bits will always be false. No other dispatch operations are defined. See Figure 31.

3.4.30 CASSETTE CONTROL

FORMAT:	OP CODE	VARIANTS	0 - Halt
	0000 0000 0010	0...7	1 - Skip
			VAR - 2,3,6,7

Perform the indicated operation on the tape cassette.

V = 0 Start Tape

1 Stop/Skip Tape (The processor also halts if it is ir TAPE

2. Stop/Skip Tape if X neq Y (The processor also halts if it is in TAPE mode.)

3 Stop/Skip Tape if FA NEQ BR (The processor also halts if it is in TAPE mode.)

4 Reserved

5 Reserved

6 Stop/Skip Tape is X = Y (The processor also halts it it is in TAPE mode.)

7 Stop/Skip Tape if FA = BR (The processor also halts if it is in TAPE mode.)

Note: All Stop Tape variants cause the tape to halt in the next available gap.

3.4.31 HALT

```
        ----------------------
FORMAT: | OP CODE            |
        | 0000 0000 0000 0001 |
        ----------------------
```

Stop execution of the micro-instructions. In RUN mode the next micro to be executed is fetched and stored in the M-register and the A-register points to the next following micro. In TAPE mode the next micro is not fetched and stored in the M-register and the HALT micro is not left in the M-register.

3.4.32 NO OPERATION

```
        ----------------------
FORMAT: | OP CODE            |
        | 0000 0000 0000 0000 |
        ----------------------
```

Skip to the next sequential instruction.

3.4.33 MONITOR

```
        ----------------------------------------
FORMAT: | OP CODE   |          VARIANTS          |
        | 0000 1001 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
        ----------------------------------------
```

Skip to the next sequential instruction.

During the time this micro-operator is executing the operator and the last two bits (0 and 1) are decoded, AND-ed with the system clock and are present in the backplane as follows:

```
MONITOR     0   True for the OP Code
MONITOR  00R0   True if last two bits are 00
MONITOR  01R0   True if last two bits are 01
MONITOR  02R0   True if last two bits are 10
MONITOR  03R0   True if last two bits are 11
```

3.4.34 NANO MOVE

FORMAT:

7	6	5	4	3	2	1	0
OP CODE				V	STOPPING	24 BIT	
0000	1010			A	SEQUENCE	NANO PORTION	
				R	NUMBER		

VAR:
0 - CONTINUE
1 - ABORT

The Nano Move micro affects the next micro to be executed. This micro is executed for one clock and its effect during this one clock is exactly like a no-operation micro as far as any action on any of the addressable registers.

However, on the next micro following this nano move micro, it will stop sequencing on the sequence number corresponding to the value represented by 3 bits through 6. When the sequence number of the micro sequencer equals this stopping sequence number, the MP-3 freezes, except for shifting of the nano-register to BR. The 24 bit naro-word portion called for in bits 0-2 is shifted to BR. If the option is to abort, then once BR contains the proper nano-bits the nano-register is cleared, and the next micro in sequence is executed. If the option is to continue, then once BR contains the proper nano-bits, the nano-register continues shifting until it reaches the original position. At that point, the micro execution resumes. In either option, the original contents of BR are lost. Table 4 gives the number of sequence steps for each micro-instruction. Table 5 shows which bits of the nano-register are transferred to BR as a function of the nano-portion code.

For a one sequence micro, there will be no execution of that micro if it was preceded by the nano-move micro and the stopping sequence number was 1. If there was a two sequence micro and the stopping sequence number was 2, then sequence 1 would be completed and sequence 2 would be examined and the bits shifted into BR.

MICRO	NUMBER OF SEQUENCES
1C	1
2C	1
3C	1
4C	1
5C	1
6C	1
7C	5
8C	1
9C	1
10C	1
11C	1
123C	1
145C	1
1D	-
2D	5
3D	1-8
4D	1
5D	1
6D	1-2
7D	1
8D	1
9D	1
10D	1
11D	5
1E	5
2E	1
3E	2-3
4E	1
5E	1
6E	1
7E	1-4
1F	1
3F	3
4F	1
5F	1
8F	1
0	1

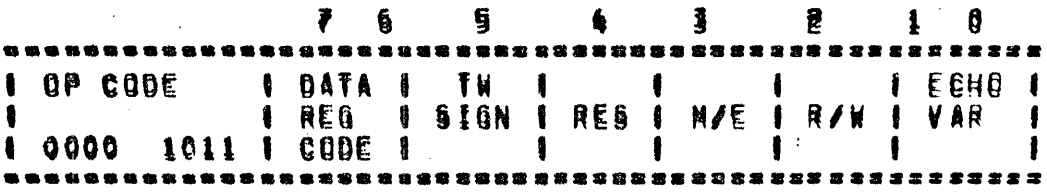
TABLE 4 - NUMBER OF SEQUENCES FOR EACH MICRO

3 2 1 0	Nano Bits
0 0 0 0	0 - 23
0 0 0 1	24 - 47
0 0 1 0	48 - 71
0 0 1 1	71 - 95
0 1 0 0	96 - 119
0 1 0 1	120 - 143
0 1 1 0	144 - 167
<	
>	Undefined
<	
>	
<	
1 1 1 1	

TABLE 5

3.4.35 DIAGNOSTIC READ/WRITE MEMORY

FORMAT:



7	6	DATA REG CODE
0	0	X=REGISTER
0	1	Y=REGISTER
1	0	T=REGISTER
1	1	L=REGISTER

5	TRANSFER WIDTH SIGN
0	+
1	-

3	MEMORY
0	MEMORY
1	ECHO

2	READ
0	READ MEMORY
1	WRITE MEMORY

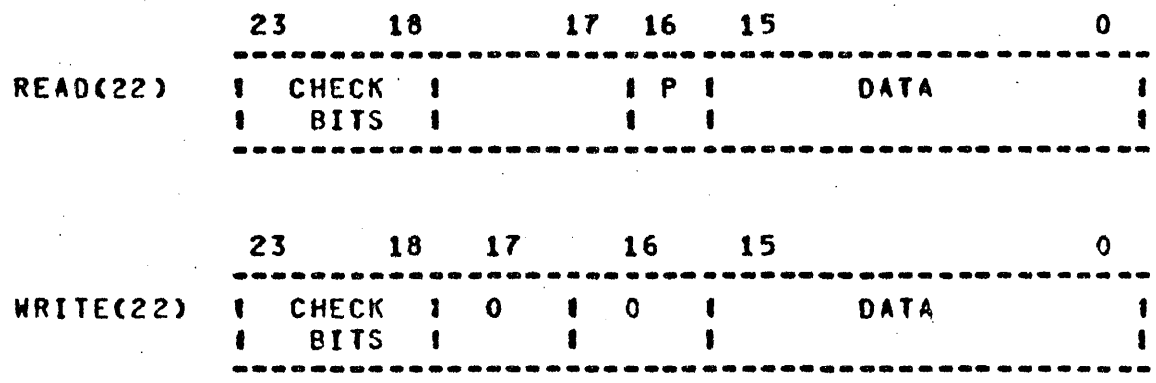
0	1	ECHO
0	0	RESERVED
0	1	WRITE DATA REGISTER
1	0	ADDRESS REGISTER
1	1	ELOG CLEAR

OPERATIONS:

A. WITH MEMORY OPTION AT MEMORY:

1) WRITE 16 + 6 (ECC) BITS TO MEMORY. Write will occur on memory stack boundaries. FA points to the memory stack. Table 4 shows the rules for generating the proper 6 bit error correcting code. See Figure 33 for the write check bit format. Write data is written from X, Y, T, or L-registers.

2) READ 16 + 6 (ECC) BITS TO X, Y, T, or L-registers. FA is the address used to determine which memory stack to read. The 16 bits are words from a memory stack and the 6 bits are the associated error correcting code bits. See Figure 33 for the code logic.



B. WITH MEMORY OPTION AT ECHO:

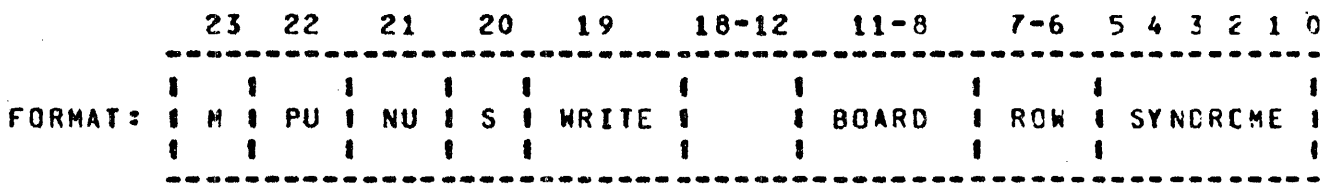
1) READ ERROR LOG REGISTER (See Figure 32) AND CLEAR THE REGISTER The ELOG register clears prior to the next load of the ELOG. Therefore consecutive reads of the ELOG register without any memory errors will give the same results.

2) READ WRITE DATA REGISTER. This will allow the processor to read back the 22-bit write data register for the key memory tibyte. The contents of this read are transferred to X, Y, T, or L-registers.

3) READ THE KEY BYTE ADDRESS REGISTER. Read the key byte address register to X, Y, T, or L-register. The transfer width sign is only used to vary the contents of the key byte adress register.

NOTE: The lower bits of FA will be ignored. This is to ensure that only memory stacks will be accessed.

ERROR REGISTER



SYNDROME — See Table 3

ROW — chip row of failure

BOARD — storage board which contain the failure

WRITE — the error described by bits 0-10 was the result of a read or write operation
WRITE = 0, READ Operation
WRITE = 1, WRITE operation

S — a single bit error which had been corrected was detected

M — there had been a duplicate of the S, NU, or PU type of error. Since the error log can only store information on one bit, error data has been lost.

NU — the uncorrectable (multiple-bit) error was not a CPU accessed error

PU — the uncorrectable error was a CPU accessed error

FIGURE 32 ERROR REGISTER FORMAT

TABLE 3. S-MEMORY SYNDROME INTERPRETATION

NO	SYNDROME PATTERN	INTERPRETATION	ACTION
0	0 0 0 0 0 0	FAILURE IN ERROR LOGIC MEMORY CIRCUIT	REPORT MULTIPLE ERROR
1	0 0 0 0 0 1	MULTIPLE ODD ERRORS	REPORT MULTIPLE ERROR
2	0 0 0 0 1 0	MULTIPLE ODD ERRORS	REPORT MULTIPLE ERROR
3	0 0 0 0 1 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
4	0 0 0 1 0 0	MULTIPLE ODD ERROR	REPORT MULTIPLE ERROR
5	0 0 0 1 0 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
6	0 0 0 1 1 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
7	0 0 0 1 1 1	MULTIPLE ODD ERROR	REPORT MULTIPLE ERROR
8	0 0 1 0 0 0	MULTIPLE ODD ERROR	REPORT MULTIPLE ERROR
9	0 0 1 0 0 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
10	0 0 1 0 1 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
11	0 0 1 0 1 1	DATA BITS, BIT 0 ERROR	CORRECT BIT 0, REPORT SINGLE ERROR
12	0 0 1 1 0 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
13	0 0 1 1 0 1	DATA BITS, BIT 1 ERROR	CORRECT BIT 1, REPORT SINGLE ERROR
14	0 0 1 1 1 0	DATA BITS, BIT 2 ERROR	CORRECT BIT 2, REPORT SINGLE ERROR
15	0 0 1 1 1 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
16	0 1 0 0 0 0	MULTIPLE ODD ERROR	REPORT MULTIPLE ERROR
17	0 1 0 0 0 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
18	0 1 0 0 1 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
19	0 1 0 0 1 1	DATA BITS, BIT 3 ERROR	CORRECT BIT 3, REPORT SINGLE ERROR
20	0 1 0 1 0 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
21	0 1 0 1 0 1	MULTIPLE ODD ERROR	REPORT MULTIPLE ERROR
22	0 1 0 1 1 0	DATA BITS, BIT 4 ERROR	CORRECT BIT 4, REPORT SINGLE ERROR
23	0 1 0 1 1 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
24	0 1 1 0 0 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
25	0 1 1 0 0 1	DATA BITS, BIT 5 ERROR	CORRECT BIT 5, REPORT SINGLE ERROR
26	0 1 1 0 1 0	DATA BITS, BIT 6 ERROR	CORRECT BIT 6, REPORT SINGLE ERROR
27	0 1 1 0 1 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
28	0 1 1 1 0 0	DATA BITS, BIT 7 ERROR	CORRECT BIT 7, REPORT SINGLE ERROR
29	0 1 1 1 0 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
30	0 1 1 1 1 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
31	0 1 1 1 1 1	CHECK BITS, BIT 1 ERROR	REPORT SINGLE ERROR

NO	SYNDROME PATTERN	INTERPRETATION	ACTION
32	1 0 0 0 0 0	MULTIPLE ODD ERROR	REPORT MULTIPLE ERROR
33	1 0 0 0 0 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
34	1 0 0 0 1 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
35	1 0 0 0 1 1	DATA BITS, BIT 8 ERROR	CORRECT BIT 8, REPORT SINGLE ERROR
36	1 0 0 1 0 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
37	1 0 0 1 0 1	DATA BITS, BIT 9 ERROR	CORRECT BIT 9, REPORT SINGLE ERROR
38	1 0 0 1 1 0	DATA BITS, BIT 10 ERROR	CORRECT BIT 10, REPORT SINGLE ERROR
39	1 0 0 1 1 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
40	1 0 1 0 0 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
41	1 0 1 0 0 1	DATA BITS, BIT 11 ERROR	CORRECT BIT 11, REPORT SINGLE ERROR
42	1 0 1 0 1 0	MULTIPLE ODD ERROR	REPORT MULTIPLE ERROR
43	1 0 1 0 1 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
44	1 0 1 1 0 0	DATA BITS, BIT 12 ERROR	CORRECT BIT 12 REPORT SINGLE ERROR
45	1 0 1 1 0 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
46	1 0 1 1 1 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
47	1 0 1 1 1 1	CHECK BITS, BIT 2 ERROR	REPORT SINGLE ERROR
48	1 1 0 0 0 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
49	1 1 0 0 0 1	DATA BITS, BIT 13 ERROR	CORRECT BIT 13, REPORT SINGLE ERROR
50	1 1 0 0 1 0	DATA BITS, BIT 14 ERROR	CORRECT BIT 14, REPORT SINGLE ERROR
51	1 1 0 0 1 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
52	1 1 0 1 0 0	DATA BITS, BIT 15 ERROR	CORRECT BIT 15, REPORT SINGLE ERROR
53	1 1 0 1 0 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
54	1 1 0 1 1 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
55	1 1 0 1 1 1	CHECK BITS, BIT 3 ERROR	REPORT SINGLE ERROR
56	1 1 1 0 0 0	MULTIPLE ODD ERROR	REPORT MULTIPLE ERROR
57	1 1 1 0 0 1	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
58	1 1 1 0 1 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
59	1 1 1 0 1 1	CHECK BITS, BIT 4 ERROR	REPORT SINGLE ERROR
60	1 1 1 1 0 0	EVEN NUMBER ERROR	REPORT MULTIPLE ERROR
61	1 1 1 1 0 1	CHECK BITS, BIT 5 ERROR	REPORT SINGLE ERROR
62	1 1 1 1 1 0	CHECK BITS, BIT 6 ERROR	REPORT SINGLE ERROR
63	1 1 1 1 1 1	NO ERROR	NO ACTION

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

RELATIONS BETWEEN CHECK BITS AND WRITE DATA

C1 - C6: Check bits; bit 1 - bit 6

X0 - X15: Write data; bit 0 - bit 15

(bit numbered as seen in MBU)

$$C1 = \{ X0 \oplus X1 \oplus X2 \oplus X3 \oplus X4 \oplus X5 \oplus X6 \oplus X7 \} /$$

$$C2 = \{ X0 \oplus X1 \oplus X2 \oplus X8 \oplus X9 \oplus X10 \oplus X11 \oplus X12 \} /$$

$$C3 = \{ X3 \oplus X4 \oplus X8 \oplus X9 \oplus X10 \oplus X13 \oplus X14 \oplus X15 \} /$$

$$C4 = \{ X0 \oplus X3 \oplus X5 \oplus X6 \oplus X8 \oplus X11 \oplus X13 \oplus X15 \} /$$

$$C5 = \{ X1 \oplus X5 \oplus X7 \oplus X9 \oplus X11 \oplus X12 \oplus X13 \oplus X15 \} /$$

$$C6 = \{ X2 \oplus X4 \oplus X6 \oplus X7 \oplus X10 \oplus X12 \oplus X14 \oplus X15 \} /$$

FIGURE 33A

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

RELATIONS BETWEEN SYNDROME BITS AND READ DATA

S1 - S6: Syndrome bits; bit 1 - bit 6

Y0 - Y21: Read data; bit 0 - bit 21

$$S1 = \{ Y0 \oplus Y1 \oplus Y2 \oplus Y3 \oplus Y4 \oplus Y5 \oplus Y6 \oplus Y7 \oplus Y16 \}$$

$$S2 = \{ Y0 \oplus Y1 \oplus Y2 \oplus Y8 \oplus Y9 \oplus Y10 \oplus Y11 \oplus Y12 \oplus Y17 \}$$

$$S3 = \{ Y3 \oplus Y4 \oplus Y8 \oplus Y9 \oplus Y10 \oplus Y13 \oplus Y14 \oplus Y15 \oplus Y18 \}$$

$$S4 = \{ Y0 \oplus Y3 \oplus Y5 \oplus Y6 \oplus Y8 \oplus Y11 \oplus Y13 \oplus Y14 \oplus Y19 \}$$

$$S5 = \{ Y1 \oplus Y5 \oplus Y7 \oplus Y9 \oplus Y11 \oplus Y12 \oplus Y13 \oplus Y15 \oplus Y20 \}$$

$$S6 = \{ Y2 \oplus Y4 \oplus Y6 \oplus Y7 \oplus Y10 \oplus Y12 \oplus Y14 \oplus Y15 \oplus Y21 \}$$

FIGURE 338

IN MBU:

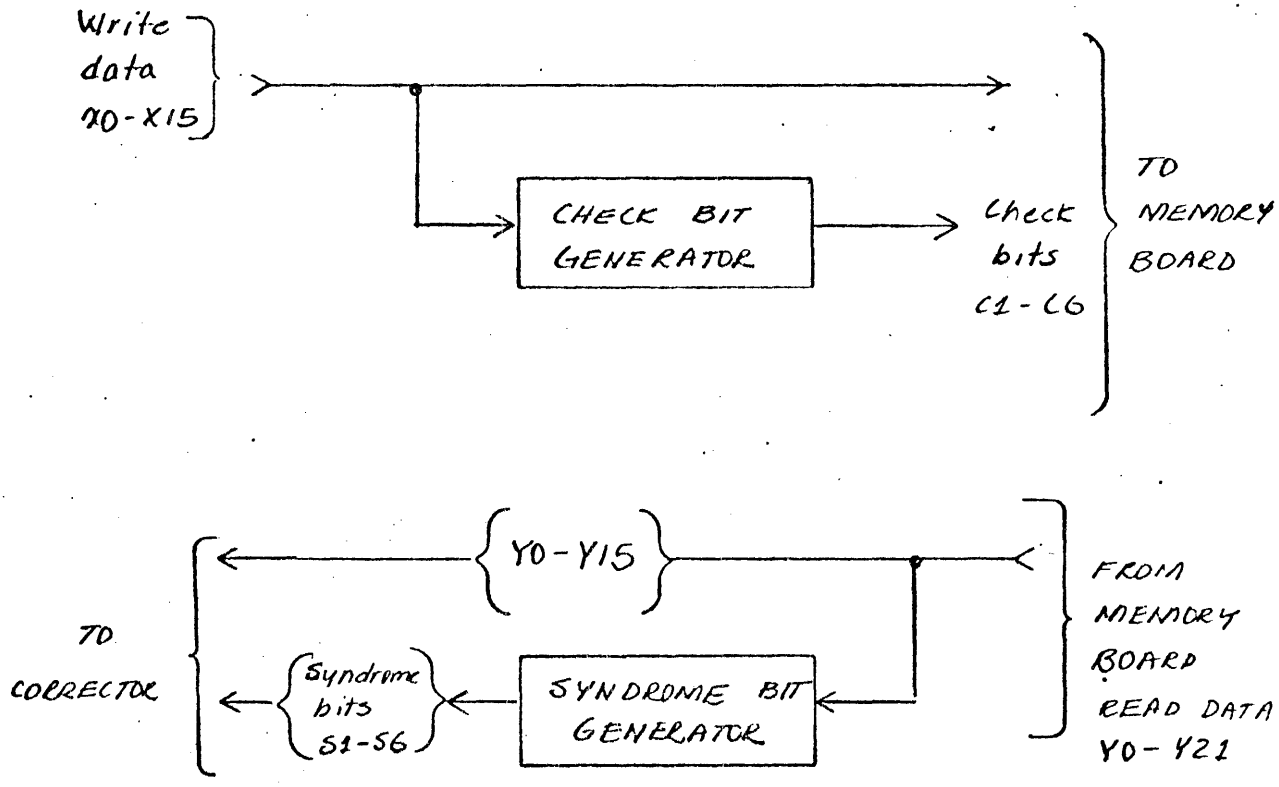


FIGURE 33C

3.4.36 CLEAR CACHE

FORMAT:

OP CODE			
0000	0000	0000	0101

This micro clears all of cache as contrasted to clearing parts of cache. In the key portion of cache, the validity bit is set tc one, data bits (8 key-bits) are set to zero, and the parity bit is set to zero. In the micro-store portion of the cache, the operation is unspecified. The LRU bits are set to indicate that micro B has been accssed; therefore the least recently used storage is micro A.

3.4.37 INCREMENT A-REGISTER

FORMAT:

0000	0000	0000	1000

This command will cause the A-register to increment by 1. In the tape mode, since the A-register does not increment after each micro, invoking this micro will cause the A-register to count up by 1.

3.5 CONTROL PANEL OPERATIONS

MCP-3 interfaces with Control Panel-4 (CP-4). CP-4 consists of three components:

a) Diagnostic and Maintenance Panel (D/M Panel),

b) Remote Operational Panel (OP Panel),

C) Remote Cassette Tape Drive (Cassette).

3.5.1 DIAGNOSTIC AND MAINTENANCE PANEL

The D/M Panel contains the following components:

a) Register Select Switch.

b) Register Group Switch.

c) M-input Switch.

d) Interrupt Switch.

e) Load Switch.

f) Halt Switch.

g) Clear Switch.

h) Start Switch.

i) Cassette Select

j) Increment Switch.

k) Single/Continuous.

l) 24 Data Switches.

m) State Light.

n) Run Light.

p) Error Light.

q) Over Temperature Light.

r) 24 Data Lights.

3.5.1.1 REGISTER SELECT SWITCH

An eight position rotary switch providing three binary encodec lines indicating switch positions. These lines are used to select the load/Display mode and the proper column of the load/Display Table. The switch positions are assigned as follows:

Position	Register Seclect Lines 4	2	1	Column Selected	Mode Selected
1	0	0	0	0	1C

2	0	0	1	1	1C
3	0	1	0	2	1C
4	0	1	1	3	1C
5	1	0	0	4	2C
6	1	0	1	5	2C
7	1	1	0	6	MEMORY
8	1	1	1	7	NOT DEFINED

The memory mode is further defined by the register group switch.

3.5.1.2 REGISTER GROUP SWITCH

A eighteen position rotary switch providing four binary encoded lines indicating switch positions. Positions 17 and 18 are interpreted the same as position 1. These Binary encoded lines are used to select the proper row in the load display table. The switch positiors are assigned as follows:

Position	8	4	2	1	Row Selected	Memory Mode
1	0	0	0	0	0	Cache Block A Write Inc 16
2	0	0	0	1	1	Cache Block B Write Inc 16
3	0	0	1	0	2	Cache Memory Read Inc 16
4	0	0	1	1	3	Cache Key Read Inc 16
5	0	1	0	0	4	Halt:Switch & A-reg equal
6	0	1	0	1	5	Halt:Switch & FA-reg (Reac) equal
7	0	1	1	0	6	Halt:Switch & FA-reg (Write) equal
8	0	1	1	1	7	Cache Clear
9	1	0	0	0	8	S-memory Read 16 Inc 16
10	1	0	0	1	9	S-memory Write 16 Inc 16
11	1	0	1	0	10	S-memory Read 22 Inc 16
12	1	0	1	1	11	S-memory Write 22 Inc 16
13	1	1	0	0	12	S-memory Read 24 Inc 24
14	1	1	0	1	13	S-memory Write 24 Inc 24
15	1	1	1	0	14	Not Defined
16	1	1	1	1	15	Read ELOG

3.5.1.3 M-INPUT SWITCH

A four position rotary switch which provides two binary encoded lines. These lines are used to select the various options for the source of instructions to be placed into the Micro-register (M) while in the Run state. The switch positions are assigned as follows:

Position	M-Input Select 2	M-Input Select 1	M-Option (Run Mode Only)
1	0	0	Enable Cache & S-memory (normal)
2	0	1	Enable S-memory
3	1	0	Enable Cache
4	1	1	LOCKED to present contents

3.5.1.4 INTERRUPT SWITCH

A two position toggle switch which provides two lines. Depending upon the position, one line is at a "one" level while the other is an open circuit. This switch is used to set the bit in the CC(0) register. It cannot reset the bit.

3.5.1.5 LOAD SWITCH

A momentary pushbutton switch which provides two lines. The lines are used to provide a pulse to determine the proper system reaction depending upon the setting of the Register select and group switches. It will cause one of the following actions:

a) load data switches into Selected Register,

b) load data switches into Selected Pad location,

c) Write data switches into cache (A),

d) Read cache (A) to console lights,

e) Write data switches to S-mem (FA),

f) Read S-mem (FA) to console lights,

g) Cause a cache clear.

h) Cause CA or RC to I/O

3.5.1.6 HALT SWITCH

A momentary pushbuttom switch used to halt the sytem while it is in the run state. The processor, upon receipt of this signal, will complete the present micro and come to an orderly halt. It will then transfer into the load/display state. In the event the system will not halt, it can be halted by depressing both clear and halt at the same

time.

3.5.1.7 CLEAR SWITCH

A momentary pushbuttom switch used to put all system registers and control F/F into the clear state. The clear signal is only active during the load/display state. If the system is in the run state, it must be halted first. In the event that the sytem will not halt, it can be cleared by depressing both clear and halt at the same time.

3.5.1.8 START SWITCH

A momentary pushbutton switch used to transfer the system from the load/display state to the RUN state. In addition, if the tape mode is selected, it will issue a cassette start signal.

3.5.1.9 CASSETTE SELECT

A two position rotary switch is used to assign the remote cassette tape drive to either the MP-3 or I/C.

3.5.1.10 INCREMENT SWITCH

A momentary pushbutton switch used, when the Memory mode has been selected, to increment the A-register (cache selected) or the FA-register (S-memory selected). The increment amount is determined by the setting of Register Group switch.

3.5.1.11 SINGLE MICRO/CONTINUOUS

A two position toggle switch used to determine run state conditions.

If this switch is in the single micro position, the system will normally be in the load/display state. When the start switch is depressed, the system will go into the run state for the execution of one micro and then return to the load/display state.

If the switch is in the continuous position, depressing the start switch will put the system into the run state. The system will serially execute instructions until requested to halt. It will then return to the load/display state.

3.5.1.12 DATA SWITCHES

24 two position toggle switches to provide data for the loading of registers or scratchpad or to provide data for a console write to either cache or S-memory. These switches may also be accessed in the run state. In the run state, these switches may be used to furnish data or provide the halt comparison address.

3.5.1.13 STATE LIGHT

A lamp which when on indicates that bit 3 of the CC-register is set.

3.5.1.14 RUN LIGHT

A lamp which when on indicates that the system is in the run state.

3.5.1.15 ERROR LIGHT

A lamp which indicates when a parity error has occurred. See Sections 3.2.10 and 3.2.11 for further details on the definition of this light.

3.5.1.16 OVER TEMPERATURE LIGHT

This lamp comes on whenever the fan is not producing acequate airflow. It indicates that the airflow is below standard, and subsequently there would be an over temperature condition.

3.5.1.17 DATA LIGHTS

There are 24 lamps that follow the main exchange of MP-3. A selected register or scratchpad location is displayed by moving it onto the main exchange. When in a memory read mode and the load button is depressed, the specified data is placed on the main exchange and locked into the data buffer for display purposes.

3.5.2 REMOTE OPERATIONAL PANEL

The Remote Operational Panel contains the following components:

a) BOT Light

b) START Switch

c) HALT INTERRUPT Switch

d) CLEAR Switch

e) TAPE/NORMAL Switch

3.5.2.1 BOT LIGHT

A lamp which indicates when the cassette tape is in the Beginring of Tape position.

3.5.2.2 START SWITCH AND LIGHT

The start switch is the same as described in Section 3.5.1.8. In addition, this switch has a lamp behind it and when it is on, it signifies a not normal run condition. The lamp is lit whenever

a) the interrupt switch is on (Section 3.5.1.4),

b) the single micro mode is selected (Section 3.5.1.11), or

c) the micro source selection is not on "normal" (Section 3.5.1.3).

3.5.2.3 INTERRUPT SWITCH AND LIGHT

This switch is used to set the CC(0) register. Whenever the CC(C) is set, the interrupt light is on.

3.5.2.4 CLEAR SWITCH

This switch has the same function as described in Section 3.5.1.7.

3.5.2.5 TAPE/NORMAL SWITCH

This switch selects whether the micros are to come from the cassette tape or from the normal source (either S-memory or cache). If the tape position is selected, then the cassette select switch (Section 3.5.1.9) must be on MP-3.

3.5.3 LOAD/DISPLAY OPERATIONS

While the sytem is in the load/display state several functiors are available at the O & M Panel. Normally the system will be responding to a display instruction until the load switch is depressed. It will then execute one load instruction and return to the cisplay instructions.

Table 3.5.3 gives the assignment of the load display furctions depending upon the positions selected for the Register Select Switch and the Register Group Switch.

REGISTER GROUP SW	REGISTER SELECT SWITCH 000	001	010	011	100	101	110	
0 0 0 0	TA	FU	X	SUM(1)	S00A	S00B	CAW(5)	1
0 0 0 1	TB	FT	Y	CMPX(1)	S01A	S02B	CBW(5)	2
0 0 1 0	TC	FLC	T	CMPY(1)	S02A	S02B	CMR(5)	3
0 0 1 1	TD	FLD	L	XANY(1)	S03A	S03B	CKR(5)	4
0 1 0 0	TE	FLE	A	XEQY(1)	S04A	S04B	SA	5
0 1 0 1	TF	FLF	M	MSKX(1)	S05A	S05B	SFAR	6
0 1 1 0	CA	BICN(1)	BF	MSKY(1)	S06A	S06B	SFAW	7
0 1 1 1	CB	FLCN(1)	LR	XORY(1)	S07A	S07B	CACLR(8)	8
1 0 0 0	LA	NULL	FA	DIFF(1)	S08A	S08B	READ16(6)	9
1 0 0 1	LB	. . .	FB	MAXS(1)	S09A	S09B	WRITE16(6)	10
1 0 1 0	LC	PERM	FL	NULL	S10A	S10B	READ22(6)	11
1 0 1 1	LD	PERP	TAS(2)	U(3)	S11A	S11B	WRITE22(6)	12
1 1 0 0	LE	XYCN(1)	CP	NULL	S12A	S12B	READ24(7)	13
1 1 0 1	LF	XYST(1)	NULL	DATA(4)	S13A	S13B	WRITE24(7)	14
1 1 1 0	CC	INCN(1)	CSW(1)	CMND(4)	S14A	S14B	. . . .	15
1 1 1 1	CD	MSSW	TIME(10)	NULL	S15A	S15B	RELOG(9)	16

NOTES

1 SOURCE ONLY

2 SOURCE ONLY PUSH OR POP OF POINTER INHIBITED WRITE INHIBITED

3 NOT AVAILABLE AS SINK OR SOURCE FROM CONSOLE

4 LOAD WILL GENERATE CA FOR DATA RC FROM CMND
RC INHIBITED when LOAD NOT USED

5 INC WILL CAUSE A TO INCREMENT BY 1(16-BIT WORD), LOAD WILL CAUSE CACHE READ OR WRITE

6 INC WILL CAUSE FA TO INCREMENT BY 16, LOAD WILL CAUSE S-MEMORY READ OR WRITE

7 INC WILL CAUSE FA TO INCREMENT BY 24, LOAD WILL CAUSE S-MEMORY READ OR WRITE

8 LOAD WILL CAUSE A CACHE CLEAR

9 LOAD WILL CAUSE S-MEMORY ELOG TO BE READ AND DISPLAYED

10 SOURCE ONLY. SINK TIME WILL CAUSE TIMER TO RESET TO ZERO.

11 The rotary has 18 positions. Position 17 and 18 are the same as 1 (i.e., Group SW - 0000).

FIGURE 34 LOAD/DISPLAY TABLE

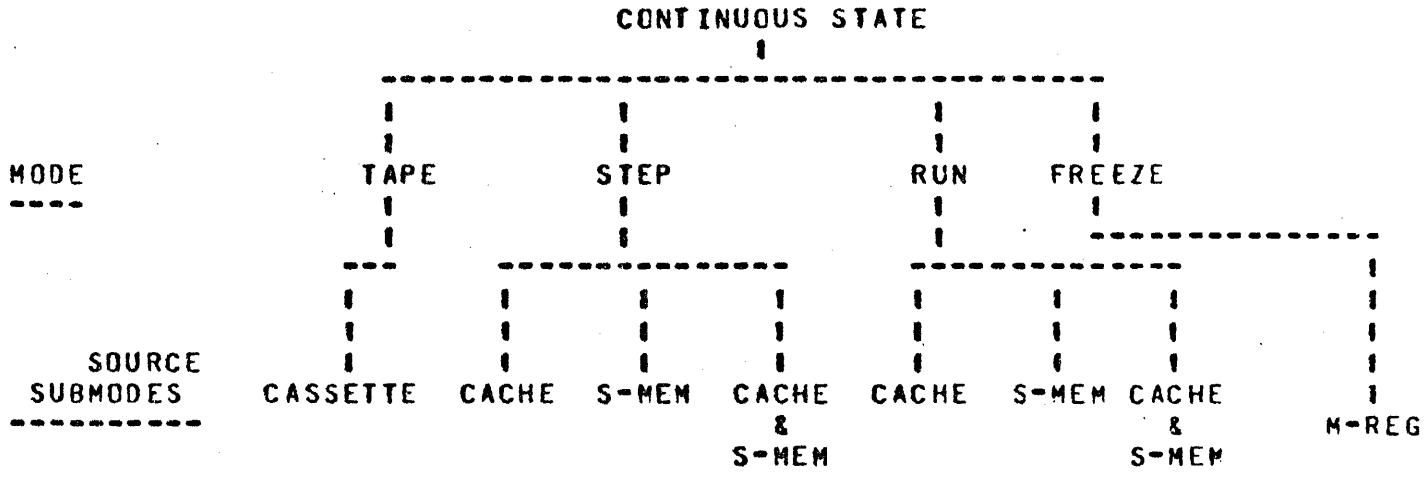
The MP-3 Processor Operations are divided into the following states, modes and sub modes.

The states are:

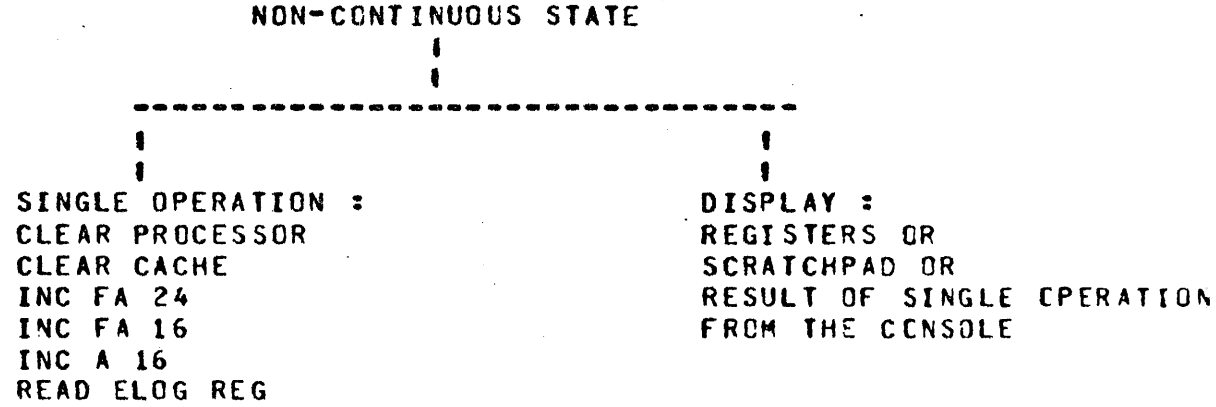
CONTINUOUS STATE: FOLLOWING PUSH OF "START" BUTTON

NON CONTINUOUS STATE: FOLLOWING PUSH OF "HALT", OR PROCESSOR SELF-INITIATED HALT

CONTINUOUS state is further divided as follows:



The non continuous state may perform any console operations; they are as follows:



WRITE 24 BIT TO MEM
WRITE 16 + 6 BIT TO MEM
READ 24 BIT FROM MEM
READ 16 + 6 BIT FROM MEM
READ/WRITE CACHE MICRO, KEY
LOAD REGISTER
LOAD SCRATCHPAD

3.6 CACHE

Cache is a high speed memory which automatically holds the most recently accessed micro-operators. In the normal operating mode, 98 percent of the time a micro is needed, it will be found in the cache. All micros are duplicated in S-memory, so that whenever a micro is not in cache, it will be automatically fetched from S-memory.

The cache organization is shown in Figure 31. Technically, the cache is of the indexed associative variety with 4 WORDS per block, 2 BLOCKS per class and 256 CLASSES. Thus, there is room for up to 2K micros.

The word to be accessed is pointed to by the low order bit of the A-register. The class is pointed to by the 8 bits of the INDEX.

The block where the micro lies is not pointed directly by the A-register. Instead, the high order bits of the A-register form the KEY. This key is associatively compared. If one of the keys match, there is a HIT. When there is a hit, the micro-operator may be fetched directly from the cache, at the word, index, and block where the hit occurred.

If neither key matches the key portion of the A-register, there is a MISS. When there is a miss, four sequential micros (64-bits) are fetched from S-memory at a location determined by the high order 18-bits of the A-register. These micros are placed in the cache at the correct index and at one of the two blocks as determined by the LRU (Least Recently Used) algorithm.

Various other capabilities are provided for normal operation and for diagnostics. A validity bit is associated with every block and says whether or not that block contains any information. The CLEAR CACHE micro and pushbutton reset all these bits (validity bit gets reset to one). Fetching and loading of micros from S-memory (on a miss) as well as writing cache by micro or from console will set the validity bit to zero.

The keys can only be viewed by the read cache key micro.

3.7 PROCESSOR CONTROL LOGIC

The Processor Control Logic consists of two main functional units: the Processor Operational Control (POC) and the Instruction Execution Control. See Figure 35.

3.8.1 INSTRUCTION EXECUTION LOGIC (IEC)

The Instruction Execution Logic accepts a micro-instruction at its input, and breaks it down to a number of suboperational steps, each of which corresponds to a nano-instruction. The nano-instruction then, is a control vector with each of its bits corresponding directly to the control points in the data structure, memory interface ard I/O interface. In addition, the sequencing-part controls the instruction decoding in four ways:

a) Parceling the nano-instruction in a predetermined sequence;

b) Allow for branching of the naro-instruction sequences;

c) Terminate a nano and supply a new nano whenever

   1. A definite clock count for the nano has been reachec, or

   2. A synchronizing signal has been received, allowirg the next suboperation;

d) Terminate the execution of a micro-instruction whenever the nano-sequence is completed, and initiate the decoding of a new micro.

The logic structure of the IEC is described in Section 3.9.

3.8.2 PROCESSOR OPERATIONAL CONTROLS

The Processor Operational Controls (POC) assumes the overall functions of coordinating the processor subunits of cache memcry, data structures, Procssor Panel Logic, Instruction Execution Control, Memory Interface, and I/O interface.

FIGURE 35

3.8.2 Cont'd

The POC performs the coordinations by controlling the micro-codes which appear as the input to the IEC. The micro-code set is the set of instructions available to the programmers plus special instructions which are needed to perform housekeeping chores within the processor. An example of some of these other instructions is loading of cache from S-memory, fetching instructions from tape, and so on.

When a micro-instruction is being executed, the micro-code that appears at the IEC input is the micro-instruction itself as it is stored in the M-reg. When a housekeeping chore is being performed, the micro-code is supplied from the Processor Operation Control ROM (POCR) of the POC.

In a HALT mode, the POC injects into the IEC the micro-coces to perform operations called for by the front panel.

In the RUN mode, the POC is responsible for loading the cache whenever a MISS is encountered, or fetching from the S-memory directly, if the cache is disabled.

In the STEP mode, the instruction in the M-register is executed. Then the M-register is loaded with the next micro. Since the machine is halted, the POC injects into the IEC the micro-codes to perform the operation as called for by the front panel.

In the TAPE mode, the POC supplies the micro-codes which results in a micro-instruction to be loaded into the M-reg from the tape. Then the micro-instruction in M is executed and the cycle is repeated.

3.9 PROCESSOR OPERATION CONTROL

A functional representation of the POC is shown in Figure 36. The console control switches are buffered and encoded to Processor Operation Control ROM (POCR). The POCR in turn, either feeds its outputs to the IEC or allows the micro-instruction from the M-REG to be executed. The rest of the logic is concerned with the phasing of the POCR operations.

The main logic conponents are treated individually.

3.9.1 CONSOLE REG SELECT ENCODERS

This unit encodes the 16 levels of the Register Row Address and the 7 levels of the Register Column Address into a 6-bit Register Address identical to the Register Address Matrix of the micro-instruction set and selection between Register Move (1C), Scratchpad move (2C), or memory operation.

These 6 lines will be used as a Source Reg Address or a Destination Reg Address in Display Update and Panel Load, respectively.

3.9.2 CONSOLE SWITCH STATES ENCODER

The console switch STATE ENCODER buffers the mode switches from the Console and further encodes them into 4 lines of processor operating modes.

In the HALT mode, it provides for the following modes:

INC A
INC FA
READ/WRITE MEMORY BY A
READ/WRITE MEMORY BY FA
LOAD
READ/WRITE CACHE FA
DISPLAY

In the other modes the following are allowed:

RUN/STEP (Cache or S-memory)
RUN/STEP (S-memory only)
RUN/STEP (Cache only)
RUN/STEP (Freeze)
TAPE

FIGURE 36

3.9.3 OPERATING MODE INTERLOCK

This part of the circuit operates in conjunction with the Console Switch State Encoder. It serves as a protection from the miscperation of the console switches and allows a gradual transition from the Halt mode to the other modes or vice versa.

In the HALT state, the console operations, except display, are initiated by the push-button of the operations. In the other states, the push-buttons are ignored.

In the RUN, STEP, or TAPE mode, any change between these modes must be punctuated first with a HALT state.

The operating Mode Interlock performs this by limiting the strobing of the latches in the Console Switch State Encoder to only (a) when the START PB is pushed, (b) in the HALT mode, after a requirec string of operation initiated by the previous PB has been completed.

3.9.4 PROCESSOR OPERATION CONTROL ROM (POCR)

The POCR outputs control the source of micro-codes to be executed by the IEC, depending upon whether it is executing a micro-instruction from a program, or performing a console-initiated operation, or performing fetches between micro-instruction executicn. The POCR derives its inputs from the Processor Operating Modes lines ard the Operating Phase Counter.

In the HALT state, the POCR supplies to the IEC the micros to perform Inc, Load, Memory Access, or register cisplay. The micro-instructions from the M-reg are ignored.

In the RUN, STEP, TAPE modes, the POCR lets the IEC execute from the M-reg and intersperses them with Load Cache, fetch from tape or fetch from memory, as they become necessary in the course of the operation.

The R[1] and R[2] are restoring elements, enabled or disabled by the POCR. When enabled, they form the source or sink reg address in a 1C micro required to perform the console display load.

The R[3] is a 16 bit restoring element, corresponding to the 16 bits in a micro-code. When it becomes necessary to load cache caused by a cache-miss or a fetch from tape, etc., the POCR supplies those commands to the IEC through R[3].

The R[4] is also a 16 bit restoring element which the POCR will enable whenever a micro from the M-reg is to be executed.

3.9.5 OPERATING PHASE COUNTER AND QUIESCENT PHASE DETECTOR

In any mode, the operation of the POCR is divided into phases. Each phase represents a stage of execution. A quiescent phase of any mode is a stage where the execution may be interrupted and a differert mode entered. Limiting the transition to the quiescent state is essential for the new mode to start at a known machine state and be terminated at a known state.

The Operating Phase Counter keeps track of the phase within a mode. In the case of executing micros from the cache, whenever a miss is encountered, the phase counter is cycled back to where the POCR will generate the Load Cache from S-memory operation. It performs in a similar manner for tape and other modes.

The Quiescent Phase Detector compares the quiescent phase number from the POCR and that of the phase counter. When there is a match, a HALT state may be entered from RUN, STEP, or TAPE. In the HALT state, a quiescent phase is when the operation initiated by the console PB is completed, and the processor is idle.

3.9.6 POC DECODE INHIBIT

This flip flop causes an all "ZERO" no-op to appear at IEC input for one clock whenever it is triggered. This is just to ensure that whenever the POCR is changed, the IOC will have one full clock to decode it.

3.9.0 INSTRUCTION EXECUTION CONTROL

This part of the control accepts the micro-code supplied by the POC, and executes it in conjunction with the cache memory, data structure, memory interface and the cassette control. The block representation is in Figure 37.

Basically, each micro-instruction is broken down into one or more steps, each a nano-instruction. The logic is considered in two parts: the part that breaks down the micro into nanos, and the part that controls the execution of the nano.

There will be a sequence number with each nano generated; the sequence number is then the step number within the micro.

The part that decodes a micro to a nano consists of a Nano Program Control Storage which derives its input addresses from the micro-instruction and the sequence number. The sequence number is derived from the sequence counter. The end of a micro is detected when the last sequence number of the micro is reached; this is detected by the sequence number comparator and specified by the MICRO END SEQ ROM.

The nano execution control logic consists of the Nano Register, the Nano Clock Counter, the Data Test Control, and the sync lines detector made up of the sync select and the slow Reg Sync Logic.

The nano word stored in the nano register consists of two parts: the Nano Program Word and the Nano Instruction Control Vector. The Nano Program Word controls the actions of the data structure, the memory interface and the I/O. The Nano Instruction Control Vector contains information lines for the number of clocks a nano will take, the sync lines to wait for, the data test conditions to look for, the delay of the sinking of a reg., etc.

FIGURE 37

3.10.1.0 MICRO TO NANO DECODE

The general operations have been discussed in the previous section. The following will deal with each subunit's implementations.

3.10.1.1.0 NANO PROGRAM CONTROL STORE (NPCS)

This is the site where all MICRO to NANO translations are stored. Each micro is translated into one or more nano words to be executed sequentially. Each nano word consists of a NANO PROGRAM WORD and NANO EXECUTION CONTROL. The Nano Program Word will be used to control the data structure and generate handshaking signals with the MBU and I/O control. The Nano Instruction Execution Control specifies any test condition to be monitored, where a slow reg may be involved, and where a nano will terminate on fixed number of clocks or on sync lines.

The NPCS is not a continuous block of PROM array but is made up of scattered sub-blocks of PROM chips. Each sub-block interprets a specific subset of the micro-instruction and for a specific subset of sequence number. For each subunit of the processor, a set of control code is required at all times to specify its behavior. The control code is derived from 5 to 10 bits of the micro-instruction. This constitutes the sub-block of PROMS. Whenever possible, more than one sub-block is combined to form PROM blocks which correspond to generating control codes for several subunits within the processor. This results in a more efficients usage of the output pins available from the PROM chips.

Although the sub-block is formed with the above criteria, some steps are still arbitrary. At the same time, such implementation results in a "RESTRICTED INPUT" PROM implementation (ie., illegal permutations are not included). This is not a handicap since the design is restricted only to implementation of the micro-set as it exists. On the other hand, it renders PROM implementation both feasible and economical in a situation which would otherwise be unfeasible.

3.10.1.1.1 NANO PROGRAM WORD

The Nano Program Word has as many subfields as there are subunits within the data structure, fetch structure, I/O interface and MBU. Each Nano Program Word selects the source register, a sink register, determines the rotations, selects the logic operations, the masking pattern, the data paths within the data structure, enabling the test circuits, enabling some registers to the MEX bus, raising requests levels to I/O or MBU whenever appropriate.

3.10.1.1.2 NANO EXECUTION CONTROL

This part contains the number of clocks the nano will take, the sync conditions necessary to terminate, the slow register delay, and the test conditions to be sensitized in detecting a SKIP, BRANCH and Branches within the nano program.

3.10.1.1.3 GENERATING ADDRESSES TO NANO PROGRAM CONTROL STORE (NPCS)

The addresses for the NPCS are derived from the micro-code that appears at the input of the IEC and the internally generated sequence number. The micro-codes are used in 3 different forms, as

a) Micro Variants and Literals,

b) Micro Selective Enables, or

c) Micro Subset Prom Addresses.

The Micro Variants and Literals are extracted directly from the micro-code without manipulations. These are fields used to specify the variants of a micro-operation, the literals to be added to some registers, and the mask to be used in testing.

The Micro Selective Enables are derived from the Micro Instruction Decoder which decodes each micro into one active line. These active lines are employed as chip enable signals for sub-blocks of PROM chips in the Nano Program Control Store.

The Micro Subset Prom Addresses are derived from the Micro Subset Encoder which derives its inputs from the outputs of the Micro Instruction Decoders. Subsets of micros that share a sub-block of PROM chips in the Nano Store are encoded together to form compressed addresses for the PROMS concerned. At the same time, they also encode the micro-set into 5 lines of addresses going into the MICRO END-SEQUENCE ROM.

The sequence number is derived from the Sequence Counter. It specifies a step number to the Nano Program Control Store during the execution of a micro-instruction. Starting at "1", it increments as each new nano word is being generated.

There are several different forms of encoded and decoded information from the micro-code that are necessary because of the Nano Programs Control Store implementation. For some sub-blocks, the encoded forms are used as addresses, and the decoded forms as the chip disable. For

some, the encoded form is not used at all, but the decoded micrc lines are used to activate some PROM chips. And in some, the encoded and decoded forms are used together as inputs to PROMS.

Similarly, the sequence number outputs are used both as chip Select Enables and chip addresses for sub-blocks of PROM chips.

3.10.1.1.4 SEQUENCE NUMBER CONTROLS

The Sequence Number Controls consist of three units: the Micro End Sequence ROM, the Sequence Number Comparator, and the Sequence Counter.

The Micro End Sequence derives its input from the 5 lines ercoded, representing 32 different micros. For each of these, it generates a 4 bit number, specifying the last sequence number for that micro.

The Sequence Number Comparator compares the Sequence Number with the last sequence number. A match will signify the end of a micro.

The Sequence Counter increments with each new nano generated, except when a terminate or nano branch condition is sensed. In the first case, it will reset to "1" with the next clock in junction with the new micro loaded in. It does this for the end of micro terminate, skip or branch, and ABORT (to be used in diagnostic routines only). When a nano branch occurs, it will backstep, ie., decrement by a fixed amount; this represents the looping within the nano program.

3.10.2.0 NANO EXECUTION CONTROL

The Nano Instruction Execution Control is made up of the Nano Register and the Nano Instruction State Control.

The nano register accepts inputs from the Nano Program Control Store. Each subfield within the nano program word is routed to the corresponding control points within the processor. The part that contains the Nano Instruction Control Vector drives the Nano State Controller directly.

3.10.2.1 NANO STATE CONTROLLER

This consists of Data Test Control, Slow Reg Sync Control, Instruction Sync Select and the Nano Clock Counter.

The Data Test Control selects the test conditions by bits from the Nano Instruction Vector. It generates 3 outputs, specifying a successful SKIP, BRANCH, or branch within the nano routine.

The Slow Reg Sync Logic delays the sinking of a register whenever a slow source is involved by specifying the Word Control Vector field. Either one or more clock is absorbed or certain sync lines being true are pre-requisite for sinking a register.

The Instruction Sync Select is used to hold over a nano instruction until a specified sync condition has arrived. This is the typical nano control for communicating with MBU and Cassette Control. It cerives the sync line selection codes from the Nano Control Vector ard the sync lines from different units in the processor.

The Nano Clock Counter is a count-down counter used to hold cver a nano instruction for a specific number (up to 15) of clocks. (This is the type of control used for communication with memory). Then, a nano instruction is executed. Beginning with the clock, it is icaced in until the the Nano Clock counter is zero. Instruction Sync Select output is true and Slow Reg inhibit is false. This allows flexibility for a nano instruction executing for a fixed number of clocks or terminating on expected conditions.

3.11 THREE PHASE DURING NORMAL RUN, TAPE AND STEP MODES

This processor is designed with a unibus which serves as the inter-register, memory, and I/O data transfers. Consequently, these operations cannot occur concurrently. However, operators not involving memory are allowed to go on concurrently with the memory refresh operation. The MP-3 processor allows processing concurrency different from its predecessors. This processor is deviced irto 3 semi-autonomous structures: the fetch structure which includes a 2K word cache memory, the control section which decoces the micro-operator into control vectors and coordinates overall CPU operations, and the data structure where data are actually manipulated. All three structures operate in parallel, each cealing with a separate phase of the micro-instruction execution.

The control structure operates its instruction decoding 1 clock ahead of the data structure, and the fetch structure fetches the micro-instruction 1 instruction ahead of the decode structure time. The overall effect is a pre-fetch, pre-decode type of architecture. The net processor time for executing a micro-instruction is equal to the time the operator is executing in the data structure. A more detailed discussion of this synchronization can be found in the following section.

Each micro-instruction is executed in 3 phases: pre-fetch, pre-decode, and execute. Consider a micro-instruction requiring n clocks, beginning at 0T + t until nT + t (T = 1 clock period, t = constant).

NORMAL MODE.

The three phases for the normal running mode (micro-instructions are fetched out of the cache memory) are as follows:

a) Fetch the next micro-instruction starts as soon as the fetched previous micro is being accepted into the M-register. The fetch access from the cache takes only 1 clock time. It is therefore ready at -1T + t, when initiated as late as -2T + t.

b) The decodes starts when the micro first appears in the M-register starting at -1T + t. It is completely occupiec with the decoding of this micro from -T + t until (n - 1)T + t.

c) The execution of a micro-instruction starts when its first nano-command enters the nano-register at 0T + t. The execution proceeds until nT + t. Consequently, the decode is always 1 clock ahead of the execution, and the fetch operation is a least 1 clock ahead of the decode. The timing relation is shown in Figure 38.

FIGURE 38 THREE PHASE TIMING

NORMAL MODE (MISS GENERATED BY CACHE)

Whenever an attempt to fetch from the micro-instruction results in a "MISS", the next micro is fetched from the memory starting at time (-1T + t) and loaded into the cache. Three different reasons are given for delaying the micro-instruction fetch until this time.

a) One full clock time is to be allowed for the cache to either supply the next micro-instruction or generate a "MISS". When the previous micro is a one clock micro, the cache canrot be expected to generate a MISS sooner.

b) The fetch will require the same bus currently used for data processing. Hence, the read request to memory will be gererated only at the end of the currect micro, assuming the bus is freed up.

c) The decode logic which is also used to decode the fetch is freed up from the current micro only at (-1T + t).

After the cache has been loaded with four successive micro-instructions, the instruction again resumes beginning at (-2T + t), as in the normal mode.

TAPE MODE

The micro-instruction is again treated in 3 phases, but the concurrency of the fetch, decode and data structure operating on different micros are eliminated.

The FETCH MICRO INSTRUCTION FROM TAPE (EMIT) will take n clocks. The fetch instruction begins its execution at t. Then the fetched micro will enter the decode network at (-1T + t). The EMIT therefore starts at -(1 = m) + T.

The timing for the Tape Mode is as follows:

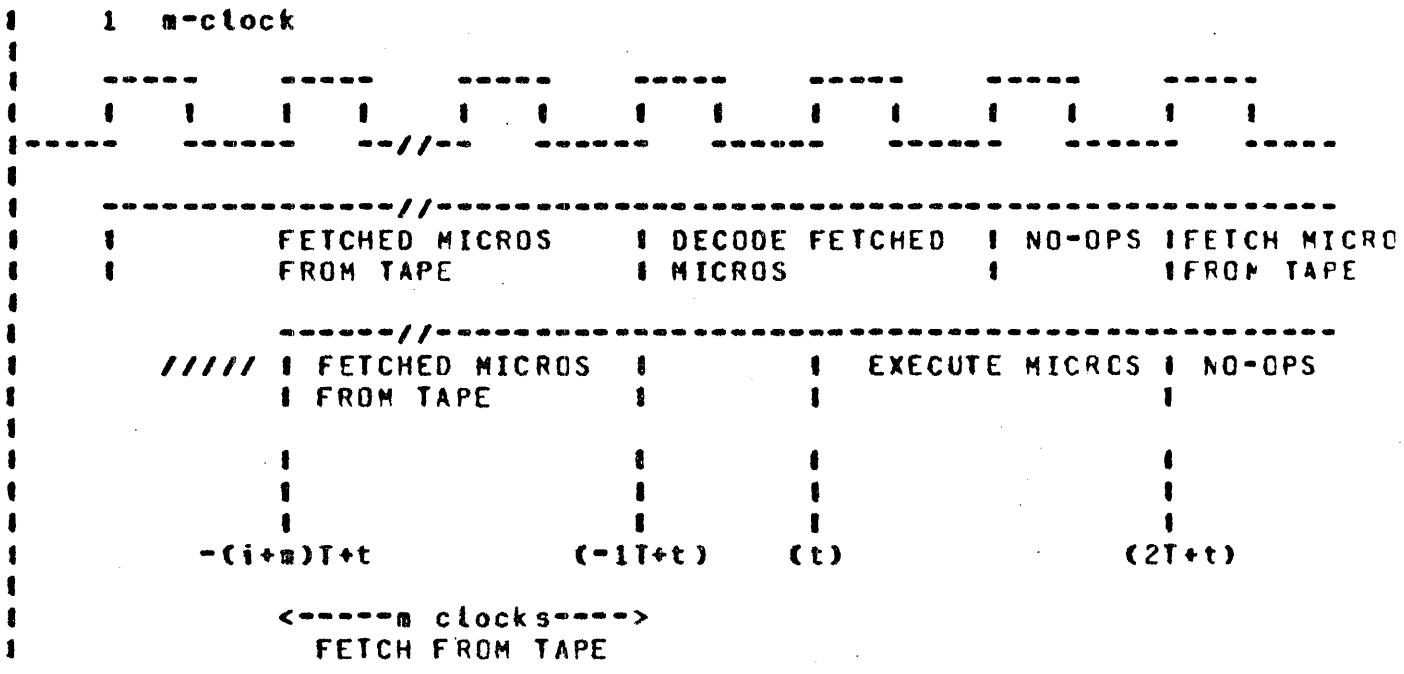


FIGURE 39

A more detailed analysis of the step-by-step transitior from execution of a micro to the fetch from tape is described in the PROCESSOR CONTROL section. SKIP

STEP MODE

In the step mode, the processor instruction timing is similar to the RUN mode. The only difference is at nT + t (the end of a micro's execution). NO-OPS are forced through the machine via the inputs to the Instruction Decoding Logic (IDL). NO-OPS are executed urtil a change in the front panel controls is effected, whereupon the processor enters the mode op operation indicated by the front panel.

The processor's micro-instruction set is implemented by execution of nano-instructions. The nano-instruction thus represents a sub-step within a micro-instruction. A concatenation of the sut-steps constitutes a micro.

The following table shows the total execution time for nT for each micro-instruction.

4.0 M-INSTRUCTION TIMING

The following instruction times (execute plus fetch of next M-Instruction) are given for the case where the next M-Instruction is contained in cache. In general, 11 clocks are added to the basic time if the M-Instruction is one of 4 fetched from S-memory (main memory) upon a cache "miss".

M-Instruction	Clocks
REGISTER MOVE EXCEPT EXIT	1 + T1 + T2
EXIT (1BA4) = Move TAS to A-REGISTER	2*
*Add one additonal clock if previous micro uses TAS as source or sink in 1C, 2C, 8C, 9C, and 10C.	
SCRATCHPAD MOVE	1 + T1 + T2*
*Add one clock if a sequence of Read after Write to Scratchpad. Add one clock if source is a binary sum or difference.	
SWAP F WITH DOUBLEPAD WORD	1
STORE F INTO DOUBLE WORD	1
LOAD F FROM DOUBLEPAD WORD	1
MOVE 8-BIT LITERAL	1 + T2
MOVE 24-BIT LITERAL	2 + T2
(OTHERS AS destination with a "miss")	15 + T2
(TAPE MODE)	U + 1
SWAP MEMORY	10
STREAM MEMORY	
Read	6 + n words
Write	4 + n words
DIAGNOSTIC TEST	
Read Error Log	6
Read 22 Bits	6
Write 22 Bits	4
WRITE MEMORY	4
READ MEMORY	6

DISPATCH LOCKOUT	Same as READ MEMORY + 1
(Skip taken)	Same as READ MEMORY + 2
DISPATCH WRITE	Same as WRITE MEMORY
DISPATCH READ	Same as READ MEMORY + 1
DISPATCH READ AND CLEAR	Same as READ MEMORY + 1
DISPATCH PORT ABSENT	1
COUNT FA/FL (Add one clock if count two registers)	1
SCRATCHPAD RELATE FA	2
EXTRACT FROM REGISTER T	1
SHIFT/ROTATE REGISTER T LEFT	1 + T2
SHIFT/ROTATE REGISTER X/Y L/R	1
SHIFT/ROTATE REGISTER XY L/R	S/R count
NORMALIZE	FL HALT: 2 + 3N MSBX HALT: 3 + 3N
READ CACHE	7
WRITE CACHE	6
CALL	2
BRANCH	2
BIAS	1 Parameter, W/C Skip 2 1 Parameter, W Skip 3 2 Parameters, W/C Skip 3 2 Parameters, W Skip 4
SET CYF	1
4-BIT MANIPULATE	1 + T1 + T2
(Skip taken)	2 + T1 + T2
NANO MOVE	1
CLEAR CACHE	257

BIT TEST BRANCH FALSE	2 + T1
(Branch Taken)	2 + T1
BIT TEST BRANCH	2 + T1
(Branch taken)	2 + T1
SKIP WHEN	1 + T1 + T2
(Skip taken)	2 + T1 + T2
CLEAR REGISTER	1 Per register to clear
BIND	2
OVERLAY M-MEMORY (Executed as no Operation)	1
CASSETTE CONTROL	1
HALT	1
NO OPERATION	1
MONITOR	1

TABLE T1 SOURCE TIMES

REGISTER NAME	NO EXTRA CLOCKS
BICN	1
FLCN	1
XYCN	1
XYST	1
SUM, BCD	1
DIFF, BDC	2
DATA	2
U	U
All others	0

TABLE T2 SINK TIMES

REGISTER NAME	NO EXTRA CLOCKS
A	2
M	1
DATA	2
CMND	2
All others	C

4.1 M-Instruction Performance

ITEM	MICRO	MICRO-COUNT	C/M	CLOCK COUNT
1	1C REGULAR	65874	1	65874
2	1C SLOW	1319	2	2638
3	1C EXIT REGULAR	29225	2	58450
4	1C EXIT SLOW	925	3	2275
5	2C READ REGULAR			
6	2C READ SLOW			
7	2C WRITE REGULAR	35846	1	35846
8	2C WRITE SLOW	3628	2	7256
9	3C REGULAR W/O SKIP	29537	1	29537
10	3C SLOW W/O SKIP	0	2	0
11	3C REGULAR WITH SKIP	176	2	352
12	3C SLOW WITH SKIP	0	3	0
13	4C REGULAR W/O BRANCH			

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
N-PROCESSOR-3
E.D.S. #2215 9891

14	4C	SLOW W/O BRANCH	3901	3	11703
15	4C	REGULAR WITH BRANCH	19279	2	38558
16	4C	SLOW WITH BRANCH	12707	3	38121
17	5C	REGULAR W/O BRANCH	9748	2	19496
18	5C	SLOW W/O BRANCH	2309	3	6927
19	5C	REGULAR WITH BRANCH	9658	2	19316
20	5C	SLOW WITH BRANCH	4087	3	12261
21	6C	REGULAR W/O SKIP	1417	1	1417
22	6C	SLOW W/O SKIP	88	2	176
23	6C	REGULAR WITH SKIP	2374	2	4748
24	6C	SLOW WITH SKIP	4	3	12
25	7C	READ	35211	6	211266
26	7C	WRITE	7070	4	28280
27	8C	REGULAR	21266	1	21266
28	8C	SLOW	0	2	0
29	9C	REGULAR	1387	2	2774
30	9C	SLOW	0	3	0
31	10C	REGULAR	14570	1	14570
32	10C	SLOW	198	2	396
33	11C	EXTRACT	30841	1	30841
34	123C	BRANCH	51747	2	103494
35	145C	CALL	19450	2	38900
36	2D	SWAP	0	10	0
37	3D	CLEAR REGULAR	1702	1	1702
38	3D	CLEAR SLOW	636	2	1272
39	4D	S/R X OR Y 1 BIT	566	1	566
40	4D	S/R X OR Y 2 BITS	1150	1	1150
41	5D	S/R X AND Y 1 BIT	234	1	234
42	5D	S/R X AND Y 2 BITS	0	2	0
43	5D	S/R X AND Y 3 BITS	49	3	147
44	6D	COUNT 1 REG	17220	1	17220
45	6D	COUNT 2 REG	74	2	148
46	7D	XCHANGE	12942	1	12942
47	8D	RELATE	11648	2	23296
48	3E	BIAS BY 1P W/O SKIP	5399	2	10798
49	3E	BIAS BY 1P WITH SKIP	1625	3	4875
50	3E	BIAS BY 2P W/O SKIP	1363	3	4089
51	3E	BIAS BY 2P WITH SKIP	653	4	2612
52	4E	STORE	4761	1	4761
53	5E	LOAD	10669	1	10669
54	6E	CARRY BINARY	0	1	0
55	6E	CARRY BCD	201	2	402
56	3F	NORM, MSBX HALT (N=15)	186	47	8742
57	3F	NORM, FL=0 HALT (N=16)	30	51	1530
58	4F	BIND	346	3	1038
59	ZERO	NOP	176	1	176
		SUMMARY	535051	1.8269	977488