AD02/11

DIAGNOSTIC

MD-11-DZADA-C

EP-DZADA-C-DL-A

OCT 1976

COPYRIGHT ©1976

FICHE 1 OF 1

digital

Made in U.S.A.

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZADA-C
PRODUCT NAME:	AD02/AD11 DIAGNOSTIC TEST
DATE CREATED:	JUNE 12,1974
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	EARL L. BOUSE

1. ABSTRACT

THIS DIAGNOSTIC TESTS AND EXERCISES THE "ADO2/AD11". THE PROGRAM IS SELF-STARTING AND WHEN LOADED WILL TYPE OUT THE PROGRAM TITLE AND THEN REQUEST THE A/D LENGTH. THE PROGRAM WILL ACCEPT A 10 TO 14 BIT UNIPOLAR OR BIPOLAR INPUT. EXAMPLE: 10(CR)* WOULD INDICATE A 10 BIT UNIPOLAR A/D; TYPING 11+(CR) WOULD INDICATE AN 11 BIT BIPOLAR A/D. A SENTANCE IS THEN TYPED GIVING THE LETTER DESIGNATORS TO BE TYPED TO RUN ANY ONE OF THE SIX (6) SEPERATE TESTS OF WHICH THIS PROGRAM IS COMPRISED. THE PROGRAM THEN TYPES A 'CR .' AND THEN WAITS IN A KEYBOARD MONITOR MODE FOR A LETTER TO BE TYPED. ALTHOUGH THESE TESTS MAY BE RUN IN ANY ORDER IT IS IMPERATIVE THAT THE 'LOGIC' TEST IS RUN FIRST AND PROVED FULLY OPERATIONAL.

THE PROGRAM IS SET UP TO GIVE THE OPERATOR AS MUCH CONTROL OVER THE PROGRAM AS POSSIBLE VIA THE TELETYPE.TYPING A '↑C' (OBTAINED VIA TYPING THE 'CNTR' AND 'C' KEYS SIMULTANEOUSLY) WHILE RUNNING ANY TEST WILL ENABLE THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR AND AWAIT A NEW LETTER DESIGNATOR TO BE TYPED. TYPING A '↑A' WHILE IN MONITOR MODE WILL ENABLE THE LETTER DESIGNATORS TO BE RETYPED.

2. REQUIREMENTS (EQUIPMENT)

A. PDP-11/05,15,20,45
B. TELETYPE
C. ADO2/AD11 ANALOG TO DIGITAL CONVERTER

3. LOADING PROCEDURE

A. USE STANDARD PROCEDURE FOR LOADING BINARY TAPES.

4. STARTING PROCEDURE

A. THE PROGRAM IS SELF STARTING WITH A RESTART ADDRESS OF '200'.
B. THE ABSOLUTE RESTART ADDRESS IS '174' IF A NEW A/D LENGTH IS TO BE ENTERED.

5. CONSOLE SWITCH SETTINGS

A. ALL SWITCHES SHOULD BE DOWN (0) WHEN THE PROGRAM IS STARTED.
B. REFER TO THE INDIVIDUAL TEST DESCRIPTIONS FOR APPLICABLE CONSOLE SWITCH SETTINGS

* TYPE 'CARRIAGE RETURN' (CR) TO TERMINATE ALL INPUT DATA.

6. AD02/AD11 ADDRESSES & FORMATS

A. A/D STATUS REGISTER (176770)

```
   15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0
ERROR      CH. ADDRESS        DONE       GAIN

ALSO
----
BIT 6 = INTERRUPT ENABLE
BIT 2 = PRIORITY REQUEST
BIT 1 = EXT.CLOCK ENABLE
BIT 0 = A/D START
```

B. A/D BUFFER (176772)

1. VARIES WITH A/D LENGTH

7. TEST DIRECTORY

TEST	SECTION
LOGIC	8.
T'NNN	9.
CALIBRATION	10.
REPEATIBILITY	11.
GAIN ACCURACY	12.
RECOVERY	13.

8. LOGIC TEST

A. THE LOGIC TEST CONSISTS OF 39 SUBTESTS WHICH CHECK OUT THE 'AD02/AD11' LOGIC. EACH TEST IS LOOPED '2048' TIMES TO TEST LOGIC RELIABILTY.

B. RESTRICTIONS

DO 'NOT' CONNECT EXTERNAL SYNC FOR THE LOGIC TEST.

C. TEST TIME

IT TAKES APPROXIMATELY '1/2' A MINUTE TO RUN THE LOGIC TEST AND THE TELETYPE BELL IS RANG AT THE END OF EACH PASS.

```
144
145                                                                    -2-
```

(8. CONT'D)

D. LOGIC ERRORS

ON ENCOUNTERING AN ERROR (DATA SWITCHES DOWN) THE ERROR ADDRESS AND THE CONTENTS OT THE A/D STATUS REGISTER ARE TYPED OUT.

1. ERROR FORMAT

ERROR ADDRESS A/D STATUS REGISTER

E. CONTROL SWITCHES (TELETYPE)

1. ↑C (CONTROL C)

TYPING A ↑C AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT THE 'LOGIC' TEST AND RETURN TO THE MONITOR.

G.

CONSOLE SWITCH SETTINGS	FUNCTION
CONSOLE SW11=0	NORMAL RUN (2048 PASSES/TEST)
CONSOLE SW11=1	SUPPRESS SUBPROGRAM ITERATIONS
CONSOLE SW13=0	PRINT ERROR MESSAGE
CONSOLE SW13=1	INHIBIT ERROR MESSAGES
CONSOLE SW14=1	RUN SCOPE MODE
CONSOLE SW14=0	INHIBIT SCOPE MODE
CONSOLE SW15=0	CONTINUE AFTER TYPING ERROR
CONSOLE SW15=1	HALT ON ERROR

9. T'NNN (LOGIC TEST AID)

A. THIS ROUTINE IS DESIGNED TO ALLOW THE OPERATOR TO LOOP ON ANY 'LOGIC SUBTEST' REGARDLESS IF THE TEST FAILS OR NOT.

B. STARTING SEQUENCE

TYPE 'TNNN (CR)' WHERE 'NNN' IS THE OCTAL ADDRESS OF THE 'TESTX' TO BE EXECUTED. THE PROGRAM WILL THEN EXECUTE THE SUBTEST AND WILL REMAIN IN A SCOPE LOOP UNTIL THE COMPUTER IS STOPPED OR A '↑C' IS TYPED TO RETURN TO THE MONITOR.

C. RESTRICTIONS

SWITCH '11' MUST BE '0' (DOWN) TO RUN THIS TEST.

202
203

10. A/D CALIBRATION TEST

A. THE 'A/D CALIBRATION' TEST IS DESIGNED TO ACCEPT AN INPUT FROM THE TELETYPE TO INDICATE THE TYPE OF SYNC (EXTERNAL OR INTERNAL TO BE USED AND THEN TAKES CONTINUOUS CONVERSIONS USING THE 'CH.' AND 'GAIN' SELECTED VIA THE CONSOLE SWITCHES. THESE SETTINGS MAY BE CHANGED AT ANY TIME. THE CONVERSION VALUE IS EITHER PLACED IN R0 AND DISPLAYED IN THE DATA LIGHTS VIA ISSUING RESETS OR PRINTED OUT ON THE TELEPRINTER.

B. STARTING SEQUENCE

1. TYPE 'C' TO RUN THE A/D CALIBRATION TEST.
2. THE TEST HEADER PLUS A REQUEST FOR A SYNC TYPE WILL THEN BE TYPED.
3. TYPE IN THE DESIRED SYNC (CR), 'I' FOR INTERNAL;'E' FOR EXTERNAL.
4. THE PROGRAM WILL RESPOND VIA TYPING A CARRIAGE RETURN-LINE FEED AND THE TEST WILL START.

C. CALIBRATION ERROR

1. THE PRINTOUT 'ERROR BIT SET' WILL OCCUR WHILE RUNNING WITH EXTERNAL SYNC IF THE SYNC FREQUENCY IS TO FAST.

D. CONTROL SWITCHES (TELETYPE)

1. ↑A (CONTROL A)

   TYPING ↑A WILL ENABLE A NEW SYNC TYPE TO BE ENTERED.

2. ↑C (CONTROL C)

   TYPING ↑C WILL CAUSE THE PROGRAM TO EXIT THE CALIBRATION TEST AND RETURN TO THE MONITOR.

E.

CONSOLE SWITCH SETTINGS	FUNCTION
SWITCHES '0-2'	CONVERSION DELAY
SWITCHES '3-4'	GAIN SELECT (1-8)
SWITCH '7=0'	DISPLAY VALUE IN R0
SWITCH '7=1'	PRINT CONVERSION VALUE
SWITCHES '9-15'	CHANNEL SELECT

11. REPEATIBILITY TEST

A. THIS TEST REQUESTS A CH.(S), GAIN AND A COUNT SPREAD OF '1-4' TO BE TYPED IN BY THE OPERATOR. A SERIES OF '512' CONVERSIONS ARE THEN TAKEN ON THE INPUT CH.(S) AT THE SELECTED GAIN. CONVERSIONS ARE THEN AVERAGED OUT AND IF THE COUNT SPREAD IS FOUND TO BE GREATER THAN REQUEST, THE RESULTS OF THE CONVERSIONS ARE TYPED OUT. A SINGLE CHANNEL OR A SERIES OF CHANNELS MAY BE TESTED VIA TYPING EITHER 'N(CR)*' TO SELECT A SINGLE CHANNEL OR 'N,N(CR)' TO TEST A SERIES OF CHANNELS.

*NOTE 'N' WOULD BE A DECIMAL NO.

B. RESTRICTIONS

1. IF A SECOND CH. IS ENTERED, IT MUST BE LARGER THAN THE FIRST CH. OR THE INPUT WILL NOT BE ACCEPTED.
2. A VOLTAGE MUST BE PROVIDED TO THE SELECTED CH. WHICH IS LESS THAN FULL SCALE FOR THE SELECTED GAIN.

C. STARTING SEQUENCE

1. TYPE 'R' TO RUN THE 'REPEATIBILITY' TEST.
2. A REQUEST IS THEN MADE FOR CH.(S) TO BE TESTED, GAIN AND COUNT SPREAD (RANGE IN WHICH ALL 512 COUNTS MUST FALL FOR THE CH. TO BE CONSIDERED ACCEPTABLE).
3. IF THE CHANNEL IS FOUND TO BE WITHIN THE SELECTED COUNT SPREAD, THE PROGRAM WILL EITHER CONTINUE TO THE NEXT CHANNEL IF SELECTED OR RETEST THE CURRENT CHANNEL.

D. CONTROL SWITCHES (TELETYPE)

1. ↑A (CONTROL A)

TYPING A ↑A WHILE THE PROGRAM IS RUNNING WILL ENABLE A NEW CH.(S), GAIN AND COUNT SPREAD TO BE SELECTED.

2. ↑C (CONTROL)

TYPING ↑C WILL CAUSE THE PROGRAM TO EXIT THE 'REPEATIBILITY' TEST AND RETURN TO THE MONITOR.

E. CONSOLE SWITCH SETTINGS FUNCTION

CONSOLE SWITCH SETTINGS	FUNCTION
CONSOLE SW 10=0	NORMAL RUN
CONSOLE SW 10=1	PRINTOUT ALL CONVERSIONS
CONSOLE SW 13=0	PRINT ERRORS
CONSOLE SW 13=1	INHIBIT ERROR PRINTOUTS

F. REPEATIBILITY ERRORS

ON ENCOUNTERING AN ERROR (CONSOLE SWITCHES DOWN) THE ERROR DATA IS TYPED OUT.

1. ERROR FORMAT

CH.	LO	AV	HI
A	B	C	D

LO	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	HI
E	F	G	H	I	J	K	L	M	N	O	P	Q

WHERE:
A=CHANNEL BEING TESTED
B=THE LOWEST READING OF THE '512' CONVERSIONS
C=THE AVERAGE READING OF THE '512' CONVERSIONS
D=THE HIGHEST READING OF THE '512' CONVERSIONS
E=NUMBER OF COUNTS 'OUT OF RANGE' LOWER THAN 5 COUNTS
F-J=NUMBER OF COUNTS IN EACH PART LOWER THAN AVERAGE.
K=NUMBER OF COUNTS AT AVERAGE OF THE '512'
L-P=NUMBER OF COUNTS IN EACH PART HIGHER THAN AVERAGE.
Q=NUMBER OF COUNTS 'OUT OF RANGE' HIGHER THAN 5 COUNTS

12. GAIN TEST

A. THE GAIN TEST IS USED TO DETERMINE THE ACCURACY OF THE 'AD02/AD11' AT DIFFERENT GAIN SETTINGS. THE TEST REQUESTS 16 SPECIFIC VOLTAGES (8 FOR A UNIPLOAR A/D) TO BE APPLIED TO THE SELECTED CHANNEL. A SERIES OF '512' CONVERSIONS ARE TAKEN FOR EVERY VOLTAGE AND APPLICABLE GAIN SETTINGS AND THE AVERAGE IS COMPARED AGAINST THE TRUE VALUE FOR THAT SPECIFIC SETTING. IF THE AVERAGE IS MORE THAN + OR -1 COUNT FROM THE TRUE VALUE IT IS CONSIDERED IN ERROR AND THE CONVERSION RESULTS ARE TYPED OUT. AFTER TESTING ALL THE VOLTAGES AT THE SPECIFIED GAIN SETTINGS A TABLE OF THE RESULTS ARE TYPED OUT. WHEN THE COMPLETE TABLE HAS BEEN TYPED THE PROGRAM WILL REQUEST A NEW CH. TO BE TESTED.

B. STARTING PROCEDURE

1. TYPE 'G' TO RUN GAIN TEST.
2. THE MESSAGE "GAIN ACCURACY TEST. SUPPLY THE FOLLOWING VOLTAGES TO THE SELECTED CH., TYPE 'CR' TO START TEST."
3. A CH. AND A SPECIFIC VOLTAGE IS THEN REQUESTED.
4. TYPE 'CR' AFTER SUPPLYING THE REQUESTED VOLTAGE.

C. CONTROL SWITCHES (TELETYPE)

1. ↑A (CONTROL A)
   TYPING A '↑A' WILL ENABLE THE GAIN TEST TO BE RESTARTED.
2. ↑C (CONTROL C)
   TYPE A '↑C' TO RETURN CONTROL TO THE MONITOR.

D. CONSOLE SWITCH SETTINGS

CONSOLE SWITCH SETTINGS	FUNCTION
CONSOLE SW13=0	PRINT GAIN ERROR
CONSOLE SW13=1	INHIBIT TYPEOUT
CONSOLE SW14=0	LOOP ON GAIN ERROR
CONSOLE SW14=1	INHIBIT ERROR LOOPING

E. GAIN ERRORS

ON ENCOUNTERING AN ERROR (CONSOLE SWITCHES SET TO '0') THE ERROR HEADER AND ERROR DATA IS TYPED OUT AND THE TEST IS LOOPED UNTIL EITHER THE CORRECT CONVERSION RESULTS ARE OBTAINED OR SW14 IS SET TO A '1' ALLOWING THE TEST TO CONTINUE.

1. ERROR FORMAT

GAIN	VOLTAGE	AVERAGE
A	B	C

WHERE:
A=GAIN SETTING
B=TRUE VOLTAGE VALUE
C=AVERAGE OF THE CONVERSION

2. GAIN CONVERSION TABLE (EXAMPLE OF AN '11' BIT BIPOLAR A/D)

GAIN	5.0000	2.5000	1.2500	0.6250	0.3125	0.1563	0.0781	0.0390
1	2000	1000	400	200	100			
2	----	2000	1000	400	200	100		
4	----	----	2000	1000	400	200	100	
8	----	----	----	2000	1000	400	200	100
1	776000	777000	777400	777600	777700			
2	------	776000	777000	777400	777600	777700		
4	------	------	776000	777000	777400	777600	777700	

```
                                        8     ------ ------ ------ 776000 777000 777400 777600 777700
                                                                    -7-
```

13. RECOVERY TEST

A. THE "RECOVERY TEST" IS DESIGNED TO DETERMINE THE RECOVERY CAPABILITY OF THE 'AD02/AD11' GAIN AMPLIFIERS. THE TEST REQUESTS FOR TWO (2) CH. AND TWO GAIN INPUTS TO BE TYPED IN. THE TEST THEN TAKES A SERIES OF SIXTEEN (16) CONVERSIONS (8 ON EACH CH.) AND THEN TYPES OUT THE '8' CONVERSION VALUES IN THE ORDER THEY WERE TAKEN ON THE SECOND CH.

B. STARTING SEQUENCE

1. TYPE 'E' TO RUN THE RECOVERY TEST.
2. A REQUEST IS THEN MADE FOR THE CH.S TO BE TESTED.
3. TYPE 'N,N (CR)' WHERE 'N' IS ANY DECIMAL CH.
4. A REQUEST FOR 'GAINS' IS THEN MADE.
5. TYPE 'N,N (CR)' WHERE 'N' WILL BE '1,2,4 OR 8.'
6. THE PROGRAM WILL THEN TAKE CONTINUOUS CONVERSIONS TYPING OUT THE CONVERSION VALUES FOR THE SECOND CH.

EXAMPLE:
CH. A XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX

WHERE:

A=TO THE SECOND CH.
X=TO THE '8' CONVERSIONS TAKEN ON THAT CH.

C. CONTROL SWITCHES (TELETYPE)

1. ↑A (CONTROL A)
   TYPING A '↑A' WILL ENABLE A NEW SET OF CH.S AND GAINS TO BE ENTERED.
2. ↑ C (CONTROL C)
   TYPING A '↑C' WILL ENABLE THE PROGRAM TO RETURN TO THE MONITOR.

D.

CONSOLE SWITCH SETTINGS	FUNCTION
CONSOLE SW 13=0	PRINT CONVERSION VALUES
CONSOLE SW 13=1	INHIBIT PRINTOUT

14. LISTING


```
%
        .TITLE AD02/AD11 DIAGNOSTIC
        .ABS
;MAINDEC-11-DZADA-C
;COPYRIGHT  1972 , 1974
;DIGITAL EQUIPMENT CORP. MAYNARD MASS. 01754
```

```
                                ;PROGRAMMER: EARL L. BOUSE

                                ;SWITCH REGISTER DEFINITIONS AND FUNCTIONS:

        100000                  SW15=100000                     ;=1, HALT ON ERROR
        040000                  SW14=40000                      ;=1, LOOP ON CURRENT TEST
        020000                  SW13=20000                      ;=1, SUPPRESS ERROR TYPEOUT
        010000                  SW12=10000
        004000                  SW11=4000                       ;=1, SUPPRESS 'SUBPROGRAM' ITERATIONS
        002000                  SW10=2000                       ;=1, FORCE TYPEOUT (REPEATIBILITY)
        001000                  SW09=1000
        000400                  SW08=400
        000200                  SW07=200
        000100                  SW06=100
        000040                  SW05=40
        000020                  SW04=20
        000010                  SW03=10
        000004                  SW02=4
        000002                  SW01=2
        000001                  SW00=1

                                ;REGISTER DEFINITIONS

        000000                  R0=%0
        000001                  R1=%1
        000002                  R2=%2
        000003                  R3=%3
        000004                  R4=%4
        000005                  R5=%5
        000006                  SP=%6
        000007                  PC=%7

                                ;GAIN EQUIVALENCE TABLE

        000000                  G1=00
        000010                  G2=10
        000020                  G4=20
        000030                  G8=30
                                ;INSTRUCTIONS DEFINITIONS

        000001                  ADCSR=%1                        ;AD STATUS REGISTER
        000004                  INTVC=%4                        ;INTERRUPT VECTOR
        005746                  PUSH1SP=5746
        005726                  POP1SP=5726
        010046                  PUSHR0=10046
        012600                  POPR0=12600
        024646                  PUSH2SP=24646
        022626                  POP2SP=22626
        000240                  NOP=240
```

```
                                ;LOAD TRAP CATCHER INTO LOC'S 0-1000
        000000                          .=0
                                .REPT   200
                                        .+2
                                        HALT                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
                                .ENDR
```

```
                000024                          .=24
000024          011534                          PWRFAL                  ;POWER FAIL HANDLER
000026          000340                          340
                000060                          .=60
000060          011030                          XTTYIN                  ;TELEPRINTER KEYBOARD ROUTINE
000062          000340                          340
                000030                          .=30
000030          011742                          EMTSRV                  ;EMT TRAP, EMT DISPATCH SERVICE
000032          000340                          340
000034          013242                          LOGERR                  ;TRAP TRAP, LOGIC ERROR TRAP
000036          000340                          340
                000174                          .=174
000174          000167  000600                  JMP     INIT            ;INITIALIZATION ADDRESS
000200          000167  001064                  JMP     MONITR          ;PROGRAM 'RESTART' ADDRESS

                                        ;TRAP EQUIVALENCE TABLE:

                104400                  ERROR=TRAP                      ;LOGIC TEST ERROR ROUTINE
                104000                  PRINT=EMT                       ;MESSAGE PRINTER ROUTINE
                104001                  DECOCT=EMT+1                    ;DECIMAL TO OCTAL CONVERSIN ROUTINE
                104002                  SCOPE=EMT+2                     ;LOGIC TEST SCOPE SUBROUTINE
                104003                  GAININ=EMT+3                    ;ROUTINE TO REQUEST GAIN FROM TTY
                104004                  CMPUTE=EMT+4                    ;A/D AVERAGING ROUTINE
                104005                  CATORIZ=EMT+5                   ;ROUTINE TO CALCULATE THE COUNT SPREAD
                104006                  BINDEC=EMT+6                    ;BINARY TO DECIMAL CONVERSION ROUTINE
                104007                  SPACE=EMT+7                     ;TYPE 'N' SPACES
                104010                  PRTOCT=EMT+10                   ;OCTAL PRINT ROUTINE
                104011                  TTYIN=EMT+11                    ;TELETYPE INPUT ROUTINE
                104012                  WAITGN=EMT+12                   ;GAIN TEST CONVERSION ROUTINE
                104013                  TAKEGN=EMT+13                   ;GAIN TEST CONVERSION ROUTINE
                104014                  PRTAVG=EMT+14                   ;GAIN AVERAGE PRINT ROUTINE
                104015                  SIXDSH=EMT+15                   ;SUBROUTINE TO TYPE OUT '6' DASHES
                104016                  TSTTKS=EMT+16                   ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
                104017                  GAINXN=EMT+17                   ;SUBROUTINE TO DECODE A GAIN FORM TTY
```

```
                                        ;TEST INITIALIZATIN ROUTINE.  PROGRAM IS SELF STARTING TO THIS ROUTINE.
                                        ;THE ROUTINE IS EXECUTED ON LOADING ONLY

        001000                                  .=1000
001000  016706  014102          INIT:   MOV     STACK,SP        ;INIT STACK POINTER=1000
001004  012777  000340  014034          MOV     #340,@PSW
001012  005737  000042                  TST     @#42            ;TEST FOR 'DDP1' MONITOR
001016  001402                          BEQ     .+6             ;CONTINUE IF NOT OUT THERE
001020  000167  000512                  JMP     RESTRT          ;IF PRESENT, RUN LOGIC TEST
001024  012777  000100  014016          MOV     #100,@TKS       ;ENABLE TTY INTERUPTS
001032  104000                          PRINT                   ;CALL MESSAGE PRINTER VIA 'EMT'
001034  013423                          TITLE                   ;TYPE PROGRAM HEADER.
001036  005067  014034          INIT1:  CLR     ADSIGN          ;BIPOLAR=0,UNIPOLAR=1
001042  104000                          PRINT
001044  013504                          MES2                    ;REQUEST THE A/D LENGTH
001046  104011                          TTYIN                   ;WAIT FOR ENTRY
001050  104001                          DECOCT                  ;CONVERT A/D LENGTH TO OCTAL
001052  022767  000012  010356          CMP     #12,BCDTAB      ;A/D LENGTH=TO 10 BITS?
001060  001004                          BNE     SIZE11          ;BRANCH IF NO
001062  012767  001000  014022          MOV     #1000,ADSIZE    ;YES, LOAD +5V VALUE
001070  000442                          BR      LDSIZE          ;BRANCH AND SET UP TABLE
001072  022767  000013  010336  SIZE11: CMP     #13,BCDTAB      ;A/D LENGTH=10 11 BITS?
001100  001004                          BNE     SIZE12          ;BRANCH IF NOT =
001102  012767  002000  014002          MOV     #2000,ADSIZE    ;YES, LOAD +5V VALUE
001110  000432                          BR      LDSIZE
001112  022767  000014  010316  SIZE12: CMP     #14,BCDTAB      ;A/D LENGTH=TO 12 BITS?
001120  001004                          BNE     SIZE13          ;BRANCH IF NOT =
001122  012767  004000  013762          MOV     #4000,ADSIZE    ;YES, LOAD +5V VALUE
001130  000422                          BR      LDSIZE
001132  022767  000015  010276  SIZE13: CMP     #15,BCDTAB      ;A/D LENGTH=TO 13 BITS?
001140  001004                          BNE     SIZE14
001142  012767  010000  013742          MOV     #10000,ADSIZE   ;LOAD 5 VOLT VAL
001150  000412                          BR      LDSIZE
001152  022767  000016  010256  SIZE14: CMP     #16,BCDTAB      ;TEST FOR 14 BITS
001160  001403                          BEQ     SIZ14A
001162  104000                          PRINT
001164  014052                          QMARK
001166  000723                          BR      INIT1           ;RETRY
001170  012767  020000  013714  SIZ14A: MOV     #20000,ADSIZE   ;LOAD +5V VALUE
001176  012700  015240          LDSIZE: MOV     #POS500,R0
001202  016767  013704  013722          MOV     ADSIZE,TEMP1
001210  016720  013716          LDPOS:  MOV     TEMP1,(R0)+
001214  006267  013712                  ASR     TEMP1
001220  022700  015252                  CMP     #NEG500,R0      ;LOADED ALL POS. VALUES?
001224  001371                          BNE     LDPOS           ;BRANCH IF NO.
001226  016767  013660  013676          MOV     ADSIZE,TEMP1
001234  005467  013672                  NEG     TEMP1           ;SET UP NEG. VALUES
001240  016720  013666          LDNEG:  MOV     TEMP1,(R0)+
001244  006267  013662                  ASR     TEMP1
001250  022700  015264                  CMP     #NEG312+2,R0    ;LOADED ALL NEG VALUES?
001254  001371                          BNE     LDNEG           ;BRANCH IF NO
001256  006367  013630                  ASL     ADSIZE          ;SET UP OFFSET FOR AVERAGING ROUTINE
001262  104000                  INITA:  PRINT
001264  013663                          MES4                    ;PRINT THE TEST CALL LETTERS.
```

```
625  001266  000420                    BR      INIT2           ;GO AND AWAIT COMMAND.
```

```
                                        ;MONITOR SUBROUTINE. ENTER VIA '↑C' OR A RESTART AT LOCATION '200'.

001270  000005                          MONITR: RESET                   ;INITIALIZE ON ENTRY
001272  016706  013610                          MOV     STACK,SP        ;RESET STACK POINTER
001276  012777  000340  013542                  MOV     #340,@PSW
001304  005737  000042                          TST     @#42            ;TEST FOR 'DDP1' MONITOR
001310  001402                                  BEQ     .+6             ;CONTINUE IF NOT OUT THERE
001312  000167  000220                          JMP     RESTRT          ;IF PRESENT, RUN LOGIC TEST
001316  012777  000100  013524                  MOV     #100,@TKS       ;ENABLE TTY INTERRUPTS
001324  104000                                  PRINT                   ;CALL MESSAGE PRINTER
001326  014036                                  CNTRLC                  ;TYPE '↑C'

001330  012767  001262  013552  INIT2:          MOV     #INITA,AVECTR   ;SET UP '↑A' VECTOR ADDRESS.
001336  012777  000100  013504                  MOV     #100,@TKS       ;ENABLE KEYBOARD INTERRUPT
001344  005067  013530                          CLR     SUBX            ;CLR SUBTEST REG. SWITCH
001350  104000                                  PRINT
001352  014047                                  DOT                     ;PRINT '.' TO INDICATE MONITOR READY
001354  104011                                  TTYIN                   ;WAIT FOR TTY ENTRY
001356  122767  000114  007706                  CMPB    #114,INBUF      ;TEST FOR 'L'
001364  001002                                  BNE     .+6             ;NOT 'L
001366  000167  000140                          JMP     LOGIC           ;YES, RUN 'LOGIC' TEST
001372  122767  000103  007672                  CMPB    #103,INBUF      ;TEST FOR 'C'
001400  001002                                  BNE     .+6             ;NOT 'C'
001402  000167  004556                          JMP     CALBRT          ;YES, RUN 'CALIBRATION' TEST
001406  122767  000122  007656                  CMPB    #122,INBUF      ;TEST FOR 'R'
001414  001002                                  BNE     .+6             ;NOT 'N'
001416  000167  004776                          JMP     REPTST          ;YES, RUN 'REPEATIBILITY' TEST
001422  122767  000107  007642                  CMPB    #107,INBUF      ;TEST FOR 'G'
001430  001002                                  BNE     .+6             ;NOT 'G'
001432  000167  005416                          JMP     GAIN            ;YES, RUN 'GAIN' TEST
001436  105767  013436                          TSTB    SUBX            ;TEST FOR SUBTEST
001442  001402                                  BEQ     .+6             ;BRANCH IF NOT SET
001444  000167  007216                          JMP     TESTX           ;OTHERWISE RUN LOGIC SUBTEST
001450  022767  000105  007614                  CMP     #105,INBUF              ;TEST FOR 'E'
001456  001002                                  BNE     .+6             ;NOT 'E'
001460  000167  007002                          JMP     RECVRY          ;YES, RUN RECOVERY TEST
001464  104000                          INIT3:  PRINT                   ;ILLEGAL ENTRY
001466  014052                                  QMARK                   ;TYPE '?'
001470  000717                                  BR      INIT2           ;WAIT AGAIN
```

```
001472  016701  013366          SETUP:  MOV     ADCS,ADCSR        ;LOAD R1 WITH ADCSR ADDRESS
001476  016704  013370                  MOV     ADINT,INTVC       ;INITIALIZE INTERRUPT ADDRESS POINTER
001502  012767  004000  011704          MOV     #4000,ICOUNT      ;INITIALIZE MAXIMUM ITERATION COUNT
001510  012767  001562  011702          MOV     #TEST0+2,RETURN   ;SETUP FIRST RETURN ADDRESS FOR SCOPE
001516  012777  000340  013322          MOV     #340,@PSW         ;SET PROCESSOR PRIORITY TO 7
001524  005067  013352                  CLR     INTFLG            ;INITIALIZE INTERRUPT FLAG
001530  000207                          RTS     %7

                                ;CHECK TO SEE THAT CSR WAS CORRECTLY INITIALIZED

001532  104000                  LOGIC:  PRINT
001534  014055                          MESS                      ;TYPE 'LOGIC' TEST
001536  004767  177730          RESTRT: JSR     %7,SETUP          ;INITIALIZE
001542  105777  013306                  TSTB    @TPS              ;WAIT FOR TTY READY
001546  100375                          BPL     .-4
001550  000005                          RESET                     ;REINITIALIZE PERIPHERALS
001552  005711                          TST     @ADCSR            ;CHECK FOR CORRECT INITIALIZATION OF CSR
001554  001401                          BEQ     .+4               ;BRANCH IF ALL ZERO
001556  104400                          ERROR                     ;CSR WAS NOT INITIALIZED TO ZERO
001560  104002                  TEST0:  SCOPE
                                ;SHOW THAT READ/WRITE BITS OF THE CSR CAN BE SET TO 0 OR 1 AND READ BACK
001562  005777  013302                  TST     @ADDBR            ;CLEAR DONE
001566  005011                          CLR     @ADCSR            ;SET TO 0
001570  032711  177577                  BIT     #177577,@ADCSR    ;CHECK ALL BITS BUT DONE
001574  001401                          BEQ     .+4               ;BRANCH IF ALL BITS READ BACK AS ZERO
001576  104400                          ERROR                     ;NOT ALL BITS READ BACK AS ZERO
001600  005777  013264                  TST     @ADDBR            ;CLEAR DONE
001604  052777  000340  013234          BIS     #340,@PSW         ;SET PROCESSOR PRIORITY AT 7 TO PREVENT INTERRUPTS
001612  012711  177777                  MOV     #177777,@ADCSR    ;LOAD ONES INTO ALL WRITEABLE BITS
001616  011102                          MOV     @ADCSR, R2        ;DONE MAY OR MAY NOT BE SET
001620  042702  000200                  BIC     #200,   R2        ;SO READ BACK CONTENTS OF THE REGISTER
001624  020227  177536                  CMP     R2,     #177536   ;BUT IGNORE DONE
001630  001401                          BEQ     .+4               ;BRANCH IF CORRECT
001632  104400                          ERROR                     ;CSR DID NOT READ BACK CORRECTLY
```

```
001634  104002                    TEST1:  SCOPE
                                  ;TEST FOR DONE BEING SET WITHIN 100 MICROSECONDS AFTER A/D START
                                  ;TEST READING DBR RESETS DONE TO ZERO
                                  ;TEST INDIRECTLY FOR DONE CLEARING A/D START (VIA SECOND A/D START)
001636  105011                            CLRB    @ADCSR          ;INITIALIZE CSR LOWER BYTE
001640  005777  013224                    TST     @ADDBR          ;READ DBR TO CLEAR DONE
001644  105711                            TSTB    @ADCSR          ;CHECK DONE BIT
001646  100001                            BPL     .+4             ;CONTINUE IF CLEARED
001650  104400                            ERROR                   ;READING DBR DIDN'T CLEAR DONE
001652  017767  013170  013234            MOV     @PSW,PROC       ;SAVE CONDITION CODES
001660  105211                            INCB    @ADCSR          ;START CONVERSION
001662  016767  013216  013230            MOV     DELAY1,COUNT    ;LOAD COUNTER FOR TIMING LOOP
001670  005267  013224                    INC     COUNT           ;LOOP FOR ABOUT 100 MICROSECONDS
001674  001375                            BNE     .-4
001676  105711                            TSTB    @ADCSR          ;CHECK DONE BIT
001700  100401                            BMI     .+4             ;CONTINUE IF SET
001702  104400                            ERROR                   ;DONE WASN'T SET WITHIN 100 MICROSECONDS.
                                                                  ;EITHER A/D DONE PREVIOUSLY FAILED TO CLEAR A/D START,
                                                                  ;OR DONE ITSELF FAILED HERE
001704  016777  013204  013134            MOV     PROC,@PSW       ;RESTORE CONDITION CODES

001712  104002                    TEST2:  SCOPE
                                  ;SHOW THAT A CLEAR INSTRUCTION DOESN'T CLEAR DONE
001714  005777  013150                    TST     @ADDBR          ;CLEAR DONE
001720  012711  000001                    MOV     #1,@ADCSR       ;INITIALIZE CONVERTER AND START CONVERSION
001724  105711                            TSTB    @ADCSR          ;WAIT FOR DONE
001726  100376                            BPL     .-2
001730  105011                            CLRB    @ADCSR          ;CLEAR LOWER BYTE
001732  105711                            TSTB    @ADCSR          ;CHECK DONE
001734  100401                            BMI     .+4             ;BRANCH IF SET
001736  104400                            ERROR                   ;DONE WAS CLEARED

001740  104002                    TEST3:  SCOPE
                                  ;DATOB TO LOWER BYTE OF CSR SHOULDN'T START CONVERSION
001742  005777  013122                    TST     @ADDBR          ;CLEAR DONE
001746  105011                            CLRB    @ADCSR          ;DATOB TO LOWER BYTE
001750  016767  013130  013142            MOV     DELAY1,COUNT    ;WAIT
001756  005267  013136                    INC     COUNT
001762  001375                            BNE     .-4
001764  105711                            TSTB    @ADCSR          ;CHECK DONE BIT
001766  100003                            BPL     .+10            ;BRANCH IF NOT SET
001770  104400                            ERROR                   ;DATOB TO LOWER BYTE CAUSED CONVERSION
001772  005777  013072                    TST     @ADDBR          ;CLEAR DONE
001776  105011                            CLRB    @ADCSR          ;REPEAT TEST, KNOWING BIT1=0
002000  016767  013100  013112            MOV     DELAY1,COUNT    ;WAIT AGAIN
002006  005267  013106                    INC     COUNT
002012  001375                            BNE     .-4
002014  105711                            TSTB    @ADCSR          ;CHECK DONE BIT
002016  100001                            BPL     .+4             ;CONTINUE IF CLEARED
002020  104400                            ERROR                   ;DATOB TO LOWER BYTE CAUSED CONVERSION
```

```
002022  104002                  TEST4:  SCOPE
                                ;NEW MUX AND GAIN VIA DATO WITH BIT1=0 SHOULD START CONVERSION
002024  005777  013040                  TST     @ADDBR          ;CLEAR DONE
002030  105011                          CLRB    @ADCSR          ;CLEAR BIT1 IN CASE PREVIOUSLY SET
002032  005011                          CLR     @ADCSR          ;START CONVERSION
002034  016767  013044  013056          MOV     DELAY1,COUNT    ;WAIT AWHILE
002042  005267  013052                  INC     COUNT
002046  001375                          BNE     .-4
002050  105711                          TSTB    @ADCSR          ;CHECK DONE BIT
002052  100401                          BMI     .+4             ;CONTINUE IF SET
002054  104400                          ERROR                   ;DONE WASN'T SET WITHIN ABOUT 100 MICROSECONDS

002056  104002                  TEST5:  SCOPE
                                ;NEW MUX AND GAIN VIA DATOB (BIT1=0) SHOULD START CONVERSION
002060  005777  013004                  TST     @ADDBR          ;CLEAR DONE
002064  105011                          CLRB    @ADCSR          ;CLEAR BIT1
002066  105077  012774                  CLRB    @ADCSO  ;START CONVERSION VIA DATOB TO UPPER BYTE
002072  016767  013006  013020          MOV     DELAY1,COUNT    ;SET UP TIMING LOOP
002100  005267  013014                  INC     COUNT           ;WAIT
002104  001375                          BNE     .-4
002106  105711                          TSTB    @ADCSR          ;CHECK DONE BIT
002110  100401                          BMI     .+4             ;CONTINUE IF SET
002112  104400                          ERROR                   ;DONE WASN'T SET WITHIN ABOUT 100 MICROSECONDS

002114  104002                  TEST6:  SCOPE
                                ;NEW MUX AND GAIN WITH BIT 1 PREVIOUSLY SET SHOULDN'T START CONVERSION
                                ;NOTE THAT THE EFFECT OF BIT1 DEPENDS ON ITS STATE BEFORE THE DATO
002116  152711  000002                  BISB    #2,@ADCSR       ;SET EXTERNAL ENABLE
002122  005777  012742                  TST     @ADDBR          ;CLEAR DONE
002126  005011                          CLR     @ADCSR          ;DATO TO CSR
002130  016767  012750  012762          MOV     DELAY1,COUNT    ;WAIT AWHILE
002136  005267  012756                  INC     COUNT
002142  001375                          BNE     .-4
002144  105711                          TSTB    @ADCSR          ;CHECK DONE BIT
002146  100001                          BPL     .+4             ;CONTINUE IF CLEARED
002150  104400                          ERROR                   ;CONVERSION OCCURRED

002152  104002                  TEST7:  SCOPE
                                ;SETTING EXTERNAL ENABLE TO ONE VIA FULL WORD DATO SHOULD CAUSE CONVERSION
                                ;IF BIT1 WAS PREVIOUSLY ZERO
002154  005777  012710                  TST     @ADDBR          ;CLEAR DONE
002160  105011                          CLRB    @ADCSR          ;CLEAR LOW BYTE OF STATUS REGISTER
002162  012711  000002                  MOV     #2,@ADCSR       ;SET EXTERNAL ENABLE
002166  016767  012712  012724          MOV     DELAY1,COUNT    ;WAIT AWHILE
002174  005267  012720                  INC     COUNT
002200  001375                          BNE     .-4
002202  142711  000002                  BICB    #2,@ADCSR       ;CLEAR EXTERNAL ENABLE
002206  105711                          TSTB    @ADCSR          ;CHECK DONE
002210  100401                          BMI     .+4             ;BRANCH IF SET
002212  104400                          ERROR                   ;CONVERSION DID NOT OCCUR
```

```
002214  104002                          TEST8:  SCOPE
                                        ;SETTING A/D START WITH BIT 1 PREVIOUSLY SET SHOULD START CONVERSION
002216  005777  012646                          TST     @ADDBR          ;CLEAR DONE
002222  112711  000002                          MOVB    #2,@ADCSR       ;SET EXTERNAL ENABLE (BIT 1)
002226  005211                                  INC     @ADCSR          ;SET A/D START
002230  016767  012650  012662                  MOV     DELAY1,COUNT    ;WAIT
002236  005267  012656                          INC     COUNT
002242  001375                                  BNE     .-4
002244  105711                                  TSTB    @ADCSR          ;CHECK DONE
002246  100401                                  BMI     .+4             ;BRANCH IF SET
002250  104400                                  ERROR                   ;CONVERSION DID NOT OCCUR

002252  104002                          TEST9:  SCOPE
                                        ;TEST FOR NO EXTERNAL CONVERSION (DURING BASIC INSTRUCTION TEST NO EXTERNAL
                                        ;SIGNAL SHOULD NORMALLY OCCUR).
002254  105011                                  CLRB    @ADCSR          ;INITIALIZE CSR LOW BYTE
002256  005777  012606                          TST     @ADDBR          ;CLEAR DONE
002262  152711  000002                          BISB    #2,@ADCSR       ;ENABLE EXTERNAL CONVERSIONS
002266  005067  012626                          CLR     COUNT           ;INITIALIZE COUNTER
002272  105267  012622                          INCB    COUNT           ;WAIT AWHILE
002276  001375                                  BNE     .-4
002300  105711                                  TSTB    @ADCSR          ;CHECK DONE
002302  100001                                  BPL     TEST10          ;CONTINUE IF NO CONVERSION OCCURRED
002304  104400                                  ERROR                   ;AN EXTERNAL TRIGGER OCCURRED

002306  104002                          TEST10: SCOPE
                                        ;TWO SUCCESSIVE DATO'S SHOULD SET THE ERROR FLAG
002310  017767  012532  012576                  MOV     @PSW,PROC       ;STORE CONDITION CODES
002316  042777  000020  012522                  BIC     #20,@PSW                ;CLEAR TRACE BIT
002324  105011                                  CLRB    @ADCSR          ;INITIALIZE CSR LOW BYTE
002326  005777  012536                          TST     @ADDBR          ;CLEAR DONE
002332  005011                                  CLR     @ADCSR          ;FIRST DATO, CLEARS ERROR FLAG
002334  005011                                  CLR     @ADCSR          ;SECOND DATO, SHOULD SET ERROR FLAG
002336  005711                                  TST     @ADCSR          ;CHECK ERROR BIT
002340  100401                                  BMI     .+4             ;CONTINUE IF ERROR WAS SET
002342  104400                                  ERROR                   ;ERROR FLAG WASN'T SET
002344  016777  012544  012474                  MOV     PROC,@PSW       ;RESTORE CONDITION CODES

002352  104002                          TEST11: SCOPE
                                        ;NEW MUX AND GAIN, FOLLOWED IMMEDIATELY BY SETTING A/D START TO 1
                                        ;SHOULD SET ERROR FLAG
002354  017767  012466  012532                  MOV     @PSW,PROC       ;STORE CONDITION CODES
002362  042777  000020  012456                  BIC     #20,@PSW                ;CLEAR TRACE BIT
002370  105711                                  TSTB    @ADCSR          ;CLEAR DONE
002372  105011                                  CLRB    @ADCSR          ;INITIALZE LOW BYTE OF CSR
002374  005011                                  CLR     @ADCSR          ;DATO TO CSR
002376  005211                                  INC     @ADCSR          ;SET A/D START
002400  005711                                  TST     @ADCSR          ;CHECK ERROR
002402  100401                                  BMI     .+4             ;BRANCH IF SET
002404  104400                                  ERROR                   ;ERROR WAS NOT SET
002406  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
002410  100376                                  BPL     .-2
002412  016777  012476  012426                  MOV     PROC,@PSW       ;RESTORE CONDITION CODES
```

```
002420  104002                    TEST12: SCOPE
                                  ;BEGINNING CONVERSION AFTER SEEING DONE SET SHOULD CAUSE AN ERROR IF
                                  ;THE PREVIOUS DATA WASN'T READ FROM THE DBR
                                  ;SINCE 2 CONSECUTIVE INCB'S WILL NOT CAUSE AN ERROR,
                                  ;THE CASE OF AN ERROR DUE TO DATA NOT READ IS DISTINGUISHED
                                  ;FROM AN ERROR DUE TO PREVIOUS CONVERSION STILL IN PROCESS
002422  017767  012420  012464            MOV     @PSW,PROC       ;STORE CONDITION CODES
002430  042777  000020  012410            BIC     #20,@PSW                ;CLEAR TRACE BIT
002436  005777  012426                    TST     @ADDBR          ;CLEAR DONE
002442  012711  000001                    MOV     #1,     @ADCSR  ;INITIALIZE CSR AND START CONVERSION
002446  105711                            TSTB    @ADCSR          ;WAIT FOR DONE
002450  100376                            BPL     .-2
002452  105211                            INCB    @ADCSR          ;START SECOND CONVERSION
002454  005711                            TST     @ADCSR          ;CHECK ERROR BIT
002456  100401                            BMI     .+4             ;CONTINUE IF ERROR FLAG SET
002460  104400                            ERROR                   ;ERROR FLAG WASN'T SET
002462  016777  012426  012356            MOV     PROC,@PSW       ;RESTORE CONDITION CODES

002470  104002                    TEST13: SCOPE
                                  ;CLEARING CSR WITHOUT READING PREVIOUS DATA SHOULD SET ERROR
002472  005777  012372                    TST     @ADDBR          ;CLEAR DONE
002476  012711  000001                    MOV     #1,     @ADCSR  ;INITIALIZE CSR AND CONVERT
002502  105711                            TSTB    @ADCSR          ;WAIT FOR DONE
002504  100376                            BPL     .-2
002506  005011                            CLR     @ADCSR          ;CLEAR WITHOUT READING DATA
002510  005711                            TST     @ADCSR          ;CHECK ERROR BIT
002512  100401                            BMI     .+4             ;CONTINUE IF ERROR FLAG SET
002514  104400                            ERROR                   ;ERROR FLAG NOT SET
002516  105711                            TSTB    @ADCSR          ;WAIT FOR DONE
002520  100376                            BPL     .-2

002522  104002                    TEST14: SCOPE
                                  ;IF DATA IN BUFFER IS READ BEFORE NEXT CONVERSION
                                  ;NO ERROR SHOULD OCCUR
002524  005777  012340                    TST     @ADDBR          ;CLEAR DONE
002530  012711  000001                    MOV     #1,     @ADCSR  ;INITIALIZE CSR AND START CONVERSION
002534  105711                            TSTB    @ADCSR          ;WAIT FOR DONE
002536  100376                            BPL     .-2
002540  005777  012324                    TST     @ADDBR          ;READ DBR
002544  105211                            INCB    @ADCSR          ;START SECOND CONVERSION
002546  005711                            TST     @ADCSR          ;CHECK ERROR BIT
002550  100001                            BPL     .+4             ;CONTINUE IF NO ERROR
002552  104400                            ERROR                   ;ERROR FLAG WAS SET
002554  105711                            TSTB    @ADCSR          ;WAIT FOR DONE
002556  100376                            BPL     .-2             ;TO ALLOW LOOPING ON TEST
```

```
002560  104002                          TEST15: SCOPE
                                        ;SETTING A/D START TO ONE TWICE CONSECUTIVELY SHOULD CAUSE
                                        ;AN ERROR
002562  017767  012260  012324                  MOV     @PSW,PROC       ;STORE CONDITION CODES
002570  042777  000020  012250                  BIC     #20,@PSW                ;CLEAR TRACE BIT
002576  005777  012266                          TST     @ADDBR          ;CLEAR DONE
002602  012711  000001                          MOV     #1,     @ADCSR  ;SET A/D START AND INITIALIZE CSR
002606  005211                                  INC     @ADCSR          ;SET A/D START AGAIN
002610  005711                                  TST     @ADCSR          ;CHECK ERROR BIT
002612  100401                                  BMI     .+4             ;CONTINUE IF NO ERROR
002614  104400                                  ERROR                   ;ERROR FLAG WAS SET
002616  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
002620  100376                                  BPL     .-2             ;TO ALLOW LOOPING ON TEST
002622  016777  012266  012216                  MOV     PROC,@PSW       ;RESTORE CONDITION CODES

002630  104002                          TEST16: SCOPE
                                        ;SETTING A/D START, FOLLOWED IMMEDIATELY BY A DATOB TO HIGH BYTE,
                                        ;SHOULD SET ERROR FLAG
002632  017767  012210  012254                  MOV     @PSW,PROC       ;STORE CONDITION CODES
002640  042777  000020  012200                  BIC     #20,@PSW                ;CLEAR TRACE BIT
002646  005777  012216                          TST     @ADDBR          ;CLEAR DONE
002652  105011                                  CLRB    @ADCSR          ;INITIALIZE LOW BYTE OF CSR
002654  105211                                  INCB    @ADCSR          ;START CONVERSION
002656  005011                                  CLR     @ADCSR          ;DATOB TO HIGH BYTE OF CSR
002660  005711                                  TST     @ADCSR          ;WHICH ALSO CLEARS PREVIOUS ERROR FLAG
002662  100401                                  BMI     .+4             ;CHECK ERROR FLAG AND BRANCH IF NOT SET
002664  104400                                  ERROR                   ;ERROR FLAG WAS SET
002666  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE TO
002670  100376                                  BPL     .-2             ;ALLOW LOOPING ON TEST
002672  016777  012216  012146                  MOV     PROC,@PSW       ;RESTORE CONDITION CODES

002700  104002                          TEST17: SCOPE
                                        ;DATO WITH BIT 1=1 SHOULDN'T CAUSE ERROR EVEN THOUGH
                                        ;PREVIOUS DATA WAS NOT READ
002702  005777  012162                          TST     @ADDBR          ;CLEAR DONE
002706  105011                                  CLRB    @ADCSR          ;INITIALIZE CSR LOW BYTE
002710  012711  000002                          MOV     #2,     @ADCSR  ;CLEAR ERROR ,SET EXTERNAL ENABLE,
002714  105711                                  TSTB    @ADCSR          ;AND INITIATE CONVERSION
002716  100376                                  BPL     .-2             ;WAIT FOR DONE
002720  005011                                  CLR     @ADCSR          ;SECOND DATO, WITHOUT READING DATA
002722  005711                                  TST     @ADCSR          ;CHECK ERROR BIT
002724  100001                                  BPL     .+4             ;BRANCH IF NOT SET
002726  104400                                  ERROR                   ;ERROR WAS SET
```

```
002730  104002                          TEST18: SCOPE
                                        ;DONE SHOULD CAUSE AN INTERRUPT IF INTERRUPT ENABLE IS SET
002732  005777  012132                          TST     @ADDBR          ;CLEAR DONE
002736  105011                                  CLRB    @ADCSR          ;INITIALIZE LOW BYTE OF CSR
002740  052777  000340  012100                  BIS     #340,@PSW       ;SET PROCESSOR PRIORITY TO LEVEL 7
002746  012714  003014                          MOV     #TINTD,@INTVC   ;SETUP RETURN FROM INTERRUPT
002752  017764  012070  000002                  MOV     @PSW,   2(INTVC)
002760  042777  000340  012060                  BIC     #340,@PSW       ;SET PROCESSOR PRIORITY TO 0
002766  012711  000001                          MOV     #1,@ADCSR       ;CLEAR ERROR AND START CONVERSION
002772  112711  000100                          MOVB    #100,@ADCSR     ;SET INTERRUPT ENABLE
002776  105711                                  TSTB    @ADCSR          ;CHECK DONE
003000  100376                                  BPL     .-2
003002  104400                                  ERROR                   ;DONE DIDN'T CAUSE AN INTERRUPT
003004  016477  000002  012034                  MOV     2(INTVC),@PSW   ;RESTORE PROCESSOR STATUS
003012  000401                                  BR      .+4             ;SKIP NEXT INSTRUCTION
003014  022626                          TINTD:  CMP     (6)+,(6)+       ;RESTORE SP
003016  105011                                  CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE
003020  012714  000132                          MOV     #132,@INTVC     ;CHANGE INTERRUPT POINTER TO
003024  005037  000132                          CLR     @#132           ;CAUSE HALT

003030  104002                          TEST19: SCOPE
                                        ;ERROR SHOULD CAUSE AN INTERRUPT IF INTERRUPT ENABLE IS SET
003032  012714  003106                          MOV     #TINTE, @INTVC  ;SETUP INTERRUPT RETURN
003036  052777  000340  012002                  BIS     #340,@PSW
003044  017764  011776  000002                  MOV     @PSW,   2(INTVC)
003052  105011                                  CLRB    @ADCSR          ;PREVENT INTERRUPT FROM OCCURRING
003054  042777  000340  011764                  BIC     #340,@PSW       ;SET PROCESSOR TO LOWEST PRIORITY
003062  005777  012002                          TST     @ADDBR          ;CLEAR DONE
003066  012711  100100                          MOV     #100100, @ADCSR ;START CONVERSION, ENABLE INTERRUPTS, SET ERROR
003072  000240                                  NOP                     ;CLOCK IN INTERRUPT
003074  052777  000340  011744                  BIS     #340,@PSW       ;RAISE PROCESSOR PRIORITY
003102  104400                                  ERROR                   ;NO INTERRUPT OCCURRED
003104  000401                                  BR      .+4
003106  022626                          TINTE:  CMP     (6)+,(6)+       ;RESTORE SP
003110  105011                                  CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE
003112  012714  000132                          MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
003116  005037  000132                          CLR     @#132           ;TO CAUSE A HALT
003122  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
003124  100376                                  BPL     .-2
```

```
 987  003126  104002                    TEST20: SCOPE
 988                                    ;NO INTERRUPT SHOULD OCCUR IF PROCESSOR IS AT LEVEL 7 (BIT 2=0)
 989  003130  052777  000340  011710            BIS     #340,   @PSW     ;SET PROCESSOR PRIORITY TO LEVEL 7
 990  003136  012714  003164                    MOV     #TINT7A, @INTVC  ;SETUP RETURN
 991  003142  017764  011700  000002            MOV     @PSW,   2(INTVC)
 992  003150  105011                            CLRB    @ADCSR           ;MAKE SURE THAT CONVERSION WILL OCCUR
 993  003152  012711  100100                    MOV     #100100,@ADCSR   ;ENABLE INTERRUPTS, SET ERROR, AND START CONVERSION
 994  003156  000240                            NOP                      ;CLOCK INTERRUPT
 995  003160  105011                            CLRB    @ADCSR           ;DISABLE INTERRUPTS
 996  003162  000402                            BR      .+6              ;CONTINUE IF NO INTERRUPT
 997  003164  104400                    TINT7A: ERROR                    ;INTERRUPT OCCURRED WITH PROCESSOR AT LEVEL 7
 998  003166  022626                            CMP     (6)+,(6)+        ;RESTORE STACK POINTER
 999  003170  105711                            TSTB    @ADCSR           ;WAIT FOR DONE
1000  003172  100376                            BPL     .-2
1001  003174  012714  000132                    MOV     #132,@INTVC      ;CHANGE INTERRUPT RETURN
1002  003200  005037  000132                    CLR     @#132            ;TO CAUSE A HALT
1003
1004
1005
1006
1007                                                .MACR   INT     STATUS,N,TESTNO
1008                                    ;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 'N' (BIT 2=0)
1009                                    TESTNO: SCOPE
1010                                            BIS     #340,   @PSW     ;SET PROCESSOR PRIORITY TO LEVEL 7
1011                                            MOV     #TINT'N'A,@INTVC;SETUP RETURN
1012                                            MOV     @PSW,2(INTVC)
1013                                            MOV     #1,     @ADCSR   ;INITIALIZE CSR
1014                                            TSTB    @ADCSR           ;WAIT FOR DONE
1015                                            BPL     .-2
1016                                            MOV     #100,   @ADCSR   ;ENABLE INTERRUPTS
1017                                            MOV     #STATUS,@PSW     ;SET PROCESSOR PRIORITY TO LEVEL 'N'
1018                                            MOV     2(INTVC),@PSW    ;SET PROCESSOR BACK TO LEVEL 7
1019                                            ERROR                    ;NO INTERRUPT OCCURRED
1020                                            BR      .+4
1021                                    TINT'N'A:       POP2SP           ;RESTORE STACK POINTER
1022                                            CLRB    @ADCSR           ;CLEAR INTERRUPT ENABLE
1023                                            MOV     #132,@INTVC      ;CHANGE INTERRUPT RETURN ADDRESS
1024                                            CLR     @#132            ;TO CAUSE A HALT
1025                                            .ENDM
```

```
1026
1027
1028
1029  003204                                  INT     300,6,TEST21
1030
1031  003276                                  INT     240,5,TEST22
1032
```

```
  1033  003370                                  INT     200,4,TEST23
  1034
  1035  003462                                  INT     140,3,TEST24
  1036
```

```
  1037  003554                                  INT     100,2,TEST25
  1038
  1039  003646                                  INT     40,1,TEST26
  1040
```

```
                                        ;FIND THE LEVEL AT WHICH BIT 2=1 CAUSES AN INTERRUPT
                                        ;PRINT OUT A MESSAGE TELLING WHERE THE INTERRUPT OCCURRED
                                        ;MAKE SURE THAT IT ALWAYS OCCURS AT THE SAME LEVEL
                                        ;FIRST SEE THAT NO INTERRUPT OCCURS WITH PROCESSOR
                                        ;AT LEVEL 7 PRIORITY
003740  104002                          TEST27: SCOPE                           ;NO INTERRUPT SHOULD OCCUR WITH PROCESSOR AT LEVEL 7
003742  052777  000340  011076                  BIS     #340,@PSW               ;SET PROCESSOR TO LEVEL 7
003750  005777  011114                          TST     @ADDBR                  ;CLEAR DONE
003754  105011                                  CLRB    @ADCSR                  ;INITIALIZE LOW BYTE OF STATUS REGISTER
003756  005011                                  CLR     @ADCSR                  ;INITIALIZE STATUS REGISTER
003760  105711                                  TSTB    @ADCSR                  ;WAIT FOR DONE
003762  100376                                  BPL     .-2
003764  012714  004006                          MOV     #TINT7B,@INTVC          ;SETUP RETURN
003770  017764  011052  000002                  MOV     @PSW,2(INTVC)           ;SETUP RETURN PROCESSOR STATUS
003776  112711  000104                          MOVB    #104,@ADCSR             ;SET INTERRUPT ENABLE AND BIT 2
004002  000240                                  NOP                             ;CLOCK INTERRUPT
004004  000402                                  BR      .+6                     ;BRANCH IF NO INTERRUPT
004006  104400                          TINT7B: ERROR                           ;INTERRUPT OCCURRED WITH PROCESSOR AT LEVEL 7
004010  022626                                  CMP     (SP)+,(SP)+             ;RESTORE SP
004012  105011                                  CLRB    @ADCSR                  ;DISABLE INTERRUPTS
004014  012714  000132                          MOV     #132,@INTVC             ;CHANGE INTERRUPT RETURN
004020  005037  000132                          CLR     @#132                   ;TO CAUSE A HALT

                                        ;MAKE SURE AN INTERRUPT CAN OCCUR WITH PROCESSOR PRIORITY AT LEVEL 0
004024  104002                          TEST28: SCOPE
004026  052777  000340  011012                  BIS     #340,@PSW               ;SET PROCESSOR TO LEVEL 7
004034  005777  011030                          TST     @ADDBR                  ;CLEAR DONE
004040  105011                                  CLRB    @ADCSR                  ;CLEAR EXTERNAL ENABLE
004042  005011                                  CLR     @ADCSR                  ;INITIALIZE STATUS REGISTER
004044  105711                                  TSTB    @ADCSR                  ;WAIT FOR DONE
004046  100376                                  BPL     .-2
004050  012714  004112                          MOV     #TINTX,@INTVC           ;SET UP RETURN
004054  017764  010766  000002                  MOV     @PSW,2(INTVC)           ;SETUP RETURN PROCESSOR STATUS
004062  042777  000340  010756                  BIC     #340,@PSW               ;SET PROCESSOR TO LEVEL 0
004070  112711  000104                          MOVB    #104,@ADCSR             ;SET INTERRUPT ENABLE AND BIT 2
004074  000240                                  NOP                             ;CLOCK IN INTERRUPT
004076  105011                                  CLRB    @ADCSR                  ;DISABLE INTERRUPTS
004100  052777  000340  010740                  BIS     #340,@PSW               ;RESTORE PROCESSOR PRIORITY TO 7
004106  104400                                  ERROR                           ;NO INTERRUPT OCCURRED
004110  000402                                  BR      .+6                     ;SKIP NEXT 2 INSTRUCTIONS
004112  105011                          TINTX:  CLRB    @ADCSR                  ;CLEAR INTERRUPT ENABLE AFTER INTERRUPT
004114  022626                                  CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
004116  012714  000132                          MOV     #132,   @INTVC          ;CHANGE INTERRUPT RETURN
004122  005037  000132                          CLR     @#132                   ;TO CAUSE A HALT
```

```
                        .MACR   INT2    TSTNO,PRP,STTS,INTLVL,NXTST
                ;TEST FOR AN INTERRUPT ON LEVEL 'INTLVL' WITH BIT2 SET
                TSTNO:  SCOPE
                        BIS     #340,@PSW       ;SET PROCESSOR PRIORITY TO 7
                        TST     @ADDBR          ;CLEAR DONE
                        CLRB    @ADCSR          ;CLEAR EXTERNAL ENABLE
                        CLR     @ADCSR          ;INITIALIZE STATUS REGISTER
                        TSTB    @ADCSR          ;WAIT FOR DONE
                        BPL     .-2
                        MOV     #TINT'PRP'B,@INTVC      ;SETUP RETURN
                        MOV     @PSW,2(INTVC)   ;SETUP RETURN PROCESSOR STATUS
                        MOV     #STTS,@PSW      ;SET PROCESSOR TO LEVEL 'PRP' PRIORITY
                        MOVB    #104,@ADCSR     ;SET INTERRUPT ENABLE AND BIT 2
                        NOP                     ;CLOCK INTERRUPT
                        CLRB    @ADCSR          ;DISABLE INTERRUPTS
                        MOV     2(INTVC),@PSW   ;RESTORE PROCESSOR STATUS
                        MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
                        CLR     @#132           ;TO CAUSE A HALT
                        BR      NXTST           ;GO TO NEXT TEST

                TINT'PRP'B:     CLRB    @ADCSR          ;DISABLE FURTHER INTERRUPTS
                        MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
                        CLR     @#132           ;TO CAUSE A HALT
                        CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
                        TST     INTFLG          ;TEST TO INHIBIT TYPE OUT
                        BNE     SET'INTLVL              ;BRANCH IF FLAG SET
                        MOV     #INTLVL,INTFLG  ;SET FLAG AND LEVEL
                        PRINT                   ;SETUP FOR PRINTOUT
                        MES6                    ;PRINT MESSAGE 'BIT2 SET CAUSES AN
                        PRTOCT                  ;PRINT LEVEL NUMBER
                        INTFLG
                        PRINT                   ;CARRIAGE RETURN, LINEFEED
                        CRLF
                SET'INTLVL:     CMP     INTFLG,#INTLVL  ;CHECK PREVIOUS LEVEL
                        BPL     NXTST           ;GO TO NEXT TEST IF INTERRUPT PREVIOUSLY AT AN EQUAL
                                                ;OR HIGHER LEVEL
                        ERROR                   ;INTERRUPT OCCURRED PREVIOUSLY ONLY AT A LOWER LEVEL
                        .ENDM
```

```
004126                                  INT2    TEST29,6,300,7,TEST30

004300                                  INT2    TEST30,5,240,6,TEST31

004452                                  INT2    TEST31,4,200,5,TEST32

004624                                  INT2    TEST32,3,140,4,TEST33

004776                                  INT2    TEST33,2,100,3,TEST34

005150                                  INT2    TEST34,1,40,2,TEST35

005322                                  INT2    TEST35,0,0,1,TEST36
                                ;NO INTERRUPT SHOULD OCCUR IF INTERRUPT ENABLE IS SET AND ERROR AND DONE ARE
                                ;BOTH CLEAR
005474  104002                  TEST36: SCOPE
005476  005777  007366                  TST     @ADDBR          ;CLEAR DONE TO PREVENT ERROR
005502  105011                          CLRB    @ADCSR          ;CLEAR EXTERNAL ENABLE
005504  005011                          CLR     @ADCSR          ;INITIALIZE STATUS REGISTER
005506  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
005510  100376                          BPL     .-2
005512  005777  007352                  TST     @ADDBR          ;CLEAR DONE
005516  012714  005562                  MOV     #NINT,@INTVC    ;SETUP RETURN IN CASE
005522  052777  000340  007316          BIS     #340,@PSW
005530  017764  007312  000002          MOV     @PSW,2(INTVC)
005536  042777  000340  007302          BIC     #340,@PSW       ;SET PROCESSOR TO LOWEST PRIORITY
005544  112711  000100                  MOVB    #100,@ADCSR     ;ENABLE INTERRUPTS
005550  000240                          NOP                     ;WINDOW TO ALLOW INTERRUPTS
005552  052777  000340  007266          BIS     #340,@PSW       ;SET PROCESSOR TO HIGHEST PRIORITY
005560  000402                          BR      .+6             ;SKIP NEXT 2 INSTRUCTIONS IF NO INTERRUPT
005562  104400                  NINT:   ERROR                   ;INTERRUPT OCCURRED
005564  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
005566  012714  000132                  MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
005572  005037  000132                  CLR     @#132           ;TO CAUSE A HALT
005576  105011                          CLRB    @ADCSR          ;DISABLE INTERRUPTS

                                ;CHECK FOR REPEATABILITY OF READING DBR AT EACH GAIN
005600  104002                  TEST37: SCOPE
005602  005777  007262                  TST     @ADDBR          ;CLEAR DONE
005606  105011                          CLRB    @ADCSR          ;CLEAR LOW BYTE
005610  005067  007260                  CLR     ADWRD2          ;LOAD INITIAL CHANNEL
005614  004767  000034                  JSR     %7,CHKGAN       ;CHECK REPEATABILITY
005620  052767  000010  007246          BIS     #10,ADWRD2      ;SET GAIN TO X2
005626  112767  000020  007240          MOVB    #20,ADWRD2      ;SET GAIN TO X4
005634  004767  000014                  JSR     %7,CHKGAN       ;CHECK REPEATABILITY
005640  042767  000030  007226          BIC     #30,ADWRD2      ;SET GAIN TO X8
005646  004767  000002                  JSR     %7,CHKGAN       ;CHECK REPEATABILITY
005652  000421                          BR      TEST38          ;GO TO NEXT TEST
005654  012767  000100  007236  CHKGAN: MOV     #100,COUNT      ;SET UP COUNTER
005662  016711  007206                  MOV     ADWRD2,@ADCSR   ;CONVERT AT DESIRED GAIN
005666  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
005670  100376                          BPL     .-2
005672  017700  007172                  MOV     @ADDBR,R0       ;LOAD DATA INTO REGISTER 0
005676  027700  007166          RELOOP: CMP     @ADDBR,R0       ;REREAD DATA AND COMPARE
005702  001401                          BEQ     .+4             ;CONTINUE IF BOTH ARE THE SAME
```

```
005704  104400                          ERROR                   ;2 READINGS OF DBR WEREN'T THE SAME
005706  005367  007206                  DEC     COUNT           ;COUNT DOWN
005712  001371                          BNE     RELOOP          ;LOOP
005714  000207                          RTS     %7              ;RETURN
005716  104002                  TEST38: SCOPE
                                ;INIT PULSE SHOULD INITIALIZE THE INTERNAL "READ" FLIP-FLOP OFF
                                ;IF THIS FAILS, THE VERY FIRST CONVERSION WILL SET ERROR
005720  012767  000002  005466          MOV     #2,ICOUNT       ;RUN TEST TWICE ONLY
005726  105777  007122                  TSTB    @TPS            ;WAIT FOR TTY READY TO AVOID
005732  100375                          BPL     .-4             ;CLOBBERING ANY OUTPUT
005734  000005                          RESET                   ;SEND OUT INIT PULSE
005736  005211                          INC     @ADCSR          ;START CONVERSION
005740  005711                          TST     @ADCSR          ;CHECK ERROR
005742  100001                          BPL     .+4             ;BRANCH IF NOT SET
005744  104400                          ERROR                   ;INTERNAL FLIP-FLOP INCORRECTLY INITIALIZED
005746  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
005750  100376                          BPL     .-2
```

```
005752  104002                          TEST39: SCOPE
                                        ;MAKE SURE THAT WITH BIT 0=0, INTERRUPTS DON'T OCCUR AT BOTH LEVELS AT  ONCE
005754  012767  004000  005432          TSTDN:  MOV     #4000,ICOUNT
005762  105777  007102                          TSTB    @ADDBR          ;CLEAR DONE
005766  105011                                  CLRB    @ADCSR          ;INITIALIZE LOW BYTE OF CSR
005770  052777  000340  007050                  BIS     #340,@PSW       ;SET PROCESSOR TO LEVEL 7
005776  012714  006044                          MOV     #TINT2,@INTVC   ;SETUP RETURN FROM INTERRUPT
006002  017764  007040  000002                  MOV     @PSW,2(INTVC)
006010  042777  000340  007030                  BIC     #340,@PSW       ;SET PROCESSOR PRIORITY TO LEVEL 0
006016  012711  000001                          MOV     #1,@ADCSR       ;CLEAR ERROR AND START CONVERSION
006022  112711  000100                          MOVB    #100,@ADCSR     ;SET INTERRUPT ENABLE
006026  105711                                  TSTB    @ADCSR          ;CHECK DONE
006030  100376                                  BPL     .-2             ;BRANCH TO RESTORE ORDER
006032  104400                                  ERROR                   ;DONE DIDN'T CAUSE AN INTERRUPT
006034  016477  000002  007004                  MOV     2(INTVC),@PSW   ;RESTORE PROCESSOR STATUS
006042  000414                                  BR      TINT3+4
006044  022626                          TINT2:  CMP     (6)+,(6)+       ;RESTORE STACK POINTER
006046  012714  006070                          MOV     #TINT3,@INTVC   ;SET UP RETURN FOR ERROR INTERRUPT
006052  005077  006770                          CLR     @PSW            ;SET PROCESSOR TO LEVEL 0
006056  000240                                  NOP                     ;ALLOW INTERRUPT
006060  016477  000002  006760                  MOV     2(INTVC),@PSW   ;RESTORE PROCESSOR PRIORITY
006066  000402                                  BR      TINT3+4         ;BRANCH AROUND HALT
006070  104400                          TINT3:  ERROR
006072  022626                                  CMP     (6)+,(6)+       ;RESTORE STACK POINTER
006074  105011                                  CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE
006076  012714  000132                          MOV     #132,@INTVC     ;CHANGE INTERRUPT POINTER TO
006102  005037  000132                          CLR     @#132           ;CAUSE A HALT
006106  104002                                  SCOPE

006110  105777  006740                          TSTB    @TPS
006114  100375                                  BPL     .-4
006116  012777  000207  006732                  MOV     #207,@TPB
006124  012767  001562  005266                  MOV     #TEST0+2,RETURN
006132  012767  004000  005254                  MOV     #4000,ICOUNT
006140  013701  000042                          MOV     @#42,R1
006144  001405                                  BEQ     LPTEST          ;IF NOT RUN UNDER 'DDP1'
006146  000005                                  RESET                   ;OTHERWISE EXIT TO MONITOR
006150  004711                                  JSR     PC,(R1)
006152  000240                                  NOP
006154  000240                                  NOP
006156  000240                                  NOP
006160  000167  173352                  LPTEST: JMP     RESTRT          ;START INSTRUCTION TESTS OVER
```

```
                                        ;*****************************************************************
                                        ;CALIBRATION ROUTINE
                                        ;*****************************************************************

                                        ;ROUTINE REQUESTS THE TYPE OF 'SYNC' TO BE USED ('I' INTERNAL OR 'E' EXTERNAL).
                                        ;THE PROGRAM THEN TAKES CONTINUOUS CONVERSIONS USING DATA SW'S9-15
                                        ;TO SELECT THE CH., SW'S 3&4 TO SELECT GAIN AND EITHER SW'S 0-2 TO SELECT DELAY
                                        ;OR SW '7' TO PRINT THE CONVERSION VALUE.

006164  012767  006204  006716  CALBRT: MOV     #CALBT1,AVECTR          ;SET UP '↑A' RESTART ADDRESS
006172  104000                          PRINT
006174  014123                          MES7                    ;TYPE TEST HEADER
006176  104000                          PRINT
006200  014177                          MES10                   ;TEXT 'SYNC I OR E'
006202  000402                          BR      CALB1A          ;WAIT FOR INPUT
006204  104000                  CALBT1: PRINT
006206  015036                          MES39                   ;TEST 'SYNC?'
006210  104011                  CALB1A: TTYIN                   ;WAIT FOR INPUT.
006212  016767  003054  006674          MOV     INBUF,PROC      ;SAVE IT IN TEMP STORAGE
006220  012767  000001  006672          MOV     #1,COUNT        ;SET UP FOR '1' CONVERSION
006226  104000                          PRINT
006230  014045                          CRLF
006232  012767  000001  005154          MOV     #1,ICOUNT       ;SETUP TO PRINT '1' VALUE
006240  117767  006616  006627  CALBT2: MOVB    @SWR0,ADWRD2+1  ;GET CH. FROM THE SW REG.
006246  000241                          CLC                     ;CLEAR THE 'C' BIT
006250  106067  006621                  RORB    ADWRD2+1        ;MOVE WORD RIGHT '1'
006254  117767  006600  006612          MOVB    @SWR,ADWRD2     ;GET GAIN FROM SW REG.
006262  142767  000347  006604          BICB    #347,ADWRD2     ;CLR UNWANTED BITS, A/D WORD COMPLETE
006270  017767  006564  006624          MOV     @SWR,KSTOR1     ;GET DELAY FROM SW REG.
006276  042767  177770  006616          BIC     #177770,KSTOR1  ;CLR UNWANTED BITS
006304  017767  006550  006612          MOV     @SWR,KSTOR2     ;SAVE ORIGINAL SWITCH SETTING.
006312  022767  000105  006574          CMP     #105,PROC       ;TEST SYNC SELECT
006320  001003                          BNE     CALB2A          ;BRANCH IF NOT 'E'
006322  052767  000002  006544          BIS     #2,ADWRD2       ;OTHERWISE ADD 'EXT' SYNC BIT TO A/D.
006330  004767  003112          CALB2A: JSR     PC,ADCNVT       ;TAKE AND STORE THE CONVERSIONS
006334  105777  006520                  TSTB    @SWR            ;TEST FOR SW7 TO PRINT
006340  100007                          BPL     CALB2B          ;BRANCH IF NOT SET.
006342  012767  015400  001574          MOV     #ADBUFF,AVGTAB  ;SET UP TO PRINT VALUE
006350  104014                          PRTAVG                  ;PRINT IT
006352  104000                          PRINT
006354  014045                          CRLF
006356  000412                          BR      CALBT4          ;TEST FOR LOOP
006360  016700  007014          CALB2B: MOV     ADBUFF,R0       ;SET UP A/D BUFFER.
006364  016767  006532  006542          MOV     KSTOR1,TEMP2    ;SET UP DELAY (RESET COUNT)
006372  104016                  CALBT3: TSTTKS                  ;TEST FOR TTY FLAG
006374  000005                          RESET
006376  005367  006532                  DEC     TEMP2           ;DECREMENT DELAY
006402  100373                          BPL     CALBT3
006404  104016                  CALBT4: TSTTKS                  ;TEST FOR KEYBOARD INTERRUPT
006406  026777  006512  006444          CMP     KSTOR2,@SWR     ;TEST IF SWITCH REGISTER HAS CHANGED
006414  001745                          BEQ     CALB2A          ;BRANCH AND TAKE NEXT CONVERSION
006416  000710                          BR      CALBT2          ;YES, COMPUTE NEW INPUT
```

```
                                        ;*****************************************************************
                                        ;REPEATIBILITY TEST
                                        ;*****************************************************************

                                        ;THIS ROUTINE TO DESIGNED TO SHOW REPEATIBILITY BY TAKING A SERIES OF
                                        ;'512' CONVERSIONS AT A SPECIFIED GAIN, AVERAGING THEM AND THEN CATORIZING
                                        ;THEM IN BINS FROM THE AVERAGE PLUS & MINUS 6 COUNTS.  THE ROUTINE
                                        ;REQUESTS FOR A CHANNEL OR CHANNELS, A GAIN AND A COUNT SPREAD TO BE TYPED
                                        ;IN VIA THE OPERATOR.  A CONTINUOUS SERIES OF CONVERSIONS ARE THEN TAKEN
                                        ;AND COMPARED AGAINST THE INPUT COUNT SPREAD.  IF ALL '512' CONVERSIONS
                                        ;ARE FOUND TO BE WITHIN THE SPREAD THE NEXT CH. IS EXERCISED OTHERWISE
                                        ;THE COUNTS ARE TYPED OUT.  SETTING SWITCH '10' TO A '1' WILL FORCE A PRINTOUT
                                        ;OF THE CH (S).

006420  012767  006432  006462  REPTST: MOV     #REPT1,AVECTR   ;SET UP CNTR 'A' VECTOR ADDRESS
006426  104000                          PRINT
006430  014250                          MES13                   ;TEXT 'REPEATIBILITY TEST'
006432  005067  006502          REPT1:  CLR     MESPRT
006436  104000                          PRINT
006440  014276                          MES14                   ;REQUEST CHANNEL (S)
006442  104011                          TTYIN                   ;WAIT FOR INPUT
006444  104001                          DECOCT                  ;CONVERT TO OCTAL
006446  016767  002764  006446          MOV     BCDTAB,KSTOR1   ;SAVE AS INTIAL CH.
006454  016767  006442  006442          MOV     KSTOR1,KSTOR2   ;ALSO SAVE AS 2ND CH. ENTRY
006462  005767  002752                  TST     BCDTAB+2        ;TEST FOR SECOND ENTRY
006466  001407                          BEQ     REPT2           ;BRANCH IF NO SECOND ENTRY
006470  026767  002744  006424          CMP     BCDTAB+2,KSTOR1 ;COMPARE ENTRY 1 TO ENTRY 2
006476  100755                          BMI     REPT1           ;BRANCH AND RESTART IF ILLEGAL
006500  016767  002734  006416          MOV     BCDTAB+2,KSTOR2 ;OTHERWISE SAVE AS SECOND CH.
006506  104003                  REPT2:  GAININ                  ;SET UP GAIN
006510  104000                          PRINT
006512  014311                          MES16                   ;TEXT 'COUNT SPREAD ?'
006514  104011                          TTYIN                   ;WAIT FOR ENTRY
006516  104001                          DECOCT                  ;DECODE TO OCTAL
006520  016767  002712  006400          MOV     BCDTAB,KSTOR3   ;SAVE IT
006526  016767  006370  006374  REPT2A: MOV     KSTOR1,KSTOR4   ;SAVE STARTING CH.
006534  104016                  REPT3:  TSTTKS                  ;TEST FOR KEYBOARD FLAG
006536  012767  001000  006354          MOV     #1000,COUNT     ;SET FOR '512' CONVERSIONS
006544  116767  006360  006323          MOVB    KSTOR4,ADWRD2+1 ;MOV SELECTED CH. TO HIGH BYTE OF ADWORD
006552  004767  002670                  JSR     PC,ADCNVT       ;TAKE THE CONVERSIONS
006556  104004                          CMPUTE                  ;AVERAGE & COMPUTE DISTRIBUTION
006560  104005                          CATORIZ
006562  032777  002000  006270          BIT     #SW10,@SWR      ;TEST DATA SW10
006570  001047                          BNE     REPT4           ;IF SET, FORCE TYPE OUT
006572  032777  020000  006260  TSTCT4: BIT     #SW13,@SWR      ;TEST FOR INHIBIT TYPEOUT
006600  001116                          BNE     REPT7           ;BRANCH IF SW SET
006602  022767  000004  006316          CMP     #4,KSTOR3       ;WAS 4 TYPED
006610  001005                          BNE     TSTCT3          ;NO, TEST FOR '3'
006612  022767  001000  006416          CMP     #1000,XSPRD4    ;TOTAL COUNTS WITHIN 4 COUNTS
006620  001033                          BNE     REPT4           ;BRANCH IF NO.
006622  000505                          BR      REPT7           ;YES, TEST NEXT CH.
```

```
006624  022767  000003  006274  TSTCT3: CMP     #3,KSTOR3        ;COUNT = TO 3
006632  001005                          BNE     TSTCT2           ;NO TEST COUNT 2
006634  022767  001000  006372          CMP     #1000,XSPRD3
006642  001022                          BNE     REPT4            ;BRANCH IF COUNT NOT WITHIN 3
006644  000474                          BR      REPT7            ;YES, TEST NEXT CH.

006646  022767  000002  006252  TSTCT2: CMP     #2,KSTOR3        ;COUNT =TO 2
006654  001005                          BNE     TSTCT1           ;NO, TEST COUNT 1
006656  022767  001000  006346          CMP     #1000,XSPRD2
006664  001011                          BNE     REPT4            ;BRANCH IF NOT WITHIN 2
006666  000463                          BR      REPT7            ;YES, TEST NEXT CH.
006670  022767  000001  006230  TSTCT1: CMP     #1,KSTOR3        ;COUNT = TO 1
006676  001004                          BNE     REPT4            ;NO, REPORT EVEN IF NOT 'O'
006700  022767  001000  006322          CMP     #1000,XSPRD1
006706  001453                          BEQ     REPT7            ;BRANCH IF TOTAL WITHIN 1 COUNT
006710  104000                  REPT4:  PRINT
006712  014045                          CRLF
006714  005767  006220                  TST     MESPRT           ;TEST IF HEADER HAS BEEN TYPED
006720  001002                          BNE     REPT5            ;BRANCH IF YES
006722  104000                          PRINT
006724  014342                          MES19                    ;TEXT 'CH. HIGH AVG. LOW'
006726  104016                  REPT5:  TSTTKS                   ;TEST FOR KEYBOARD INTERRUPT
006730  104000                          PRINT
006732  014045                          CRLF                     ;CARRIAGE RETURN, LINE FEED
006734  016702  006170                  MOV     KSTOR4,R2        ;MOV. CH.
006740  104006                          BINDEC                   ;CONVERT TO DECIMAL AND PRINT
006742  104007                          SPACE
006744  104010                          PRTOCT                   ;PRINT LOW VALUE
006746  015146                          LOW
006750  104007                          SPACE
006752  104010                          PRTOCT                   ;PRINT AVERAGE VALUE
006754  015162                          AVRAGE
006756  104007                          SPACE
006760  104010                          PRTOCT                   ;PRINT HIGH VALUE
006762  015144                          HIGH
006764  005767  006150                  TST     MESPRT
006770  001002                          BNE     REPT6
006772  104000                          PRINT
006774  014374                          MES20                    ;PRINT 'COUNT SPREAD' HEADER
006776  052767  000007  006134  REPT6:  BIS     #7,MESPRT        ;INHIBIT OTHER HEADERS
007004  022767  001000  006200          CMP     #1000,AVGCNT     ;TEST IF ALL COUNTS WERE AT AVG.
007012  001411                          BEQ     REPT7            ;BRANCH TO NEXT CH. IF YES.
007014  104000                          PRINT
007016  014045                          CRLF
007020  012704  015176                  MOV     #ORLOW,R4
007024  012402                  REPT6A: MOV     (R4)+,R2
007026  104006                          BINDEC                   ;TYPE OUT COUNT SPREAD
007030  022704  015230                  CMP     #XSPRD1,R4       ;TEST FOR DONE
007034  001373                          BNE     REPT6A           ;BRANCH IF NO AND TYPE NEXT COUNT
007036  005267  006066          REPT7:  INC     KSTOR4           ;INCREMENT 'CH.'
007042  026767  006056  006060          CMP     KSTOR2,KSTOR4    ;TESTED ALL CH.(S)?
007050  002231                          BGE     REPT3            ;BRANCH IF NO AND TEST NEXT CH.
007052  000625                          BR      REPT2A           ;OTHERWISE RESET AND REPEAT
```

```
                                        ;*****************************************************************
                                        ;GAIN ACCURACY TEST
                                        ;*****************************************************************

                                        ;THE GAIN ACCURACY TEST REQUESTS FOR A DECIMAL CH. TO BE TYPED IN VIA
                                        ;THE OPERATOR.  IT THEN REQUESTS '16' SPECIFIED VOLTAGES TO BE APPLIED TO
                                        ;THAT CHANNEL.  AFTER SUPPLY THE REQUESTED VOLTAGE, THE OPERATOR TYPES A
                                        ;SPACE ON THE TELETYPE AND A SERIES OF '512' CONVERSIONS ARE THEN TAKEN
                                        ;AT SPECIFIED GAIN SETTINGS AND AVERAGED OUT.  IF THE AVERAGED VALUE
                                        ;IS FOUND TO BE MORE THAN + OR -1 COUNT FROM THE INPUT VOLTAGES TRUE
                                        ;VALUE, IT IS CONSIDERED IN ERROR AND THE ERROR VOLTAGE, THE EXPECTED VALUE
                                        ;AND THE GAIN ARE TYPED OUT.  AFTER COMPUTING INPUT VOLTAGE AVERAGES, A
                                        ;TABLE OF ALL CONVERSION RESULTS ARE TYPED OUT.

007054  012767  007066  006026  GAIN:   MOV     #GTO,AVECTR     ;SET UP '↑A' RETURN ADDRESS
007062  104000                          PRINT
007064  013522                          MES3                    ;TEXT GAIN ACCURACY TEST
007066  104000                  GTO:    PRINT
007070  014502                          MES22                   ;TEXT 'CH.?'
007072  104011                          TTYIN                   ;WAIT FOR REPLY
007074  104001                          DECOCT                  ;CONVERT TO OCTAL
007076  116767  002334  005771          MOVB    BCDTAB,ADWRD2+1 ;SET UP SELECTED CH.
007104  005067  006030                  CLR     MESPRT          ;CLR PRINT INHIBIT SWITCH

                                        ;TEST +5.0V X G1
007110  104000                  GT1:    PRINT
007112  014512                          MES23                   ;TEXT '+5.00V'
007114  104012                          WAITGN                  ;CALL THE GAIN 'WAIT' HANDLER
007116  000000                          G1                      ;GAIN X1
007120  015240                          POS500
007122  015264                          GP50X1                  ;SAVE VALUE
007124  005767  005746                  TST     ADSIGN
007130  001406                          BEQ     GT2             ;BRANCH TO NEXT TEST IF UNIPOLAR.

                                        ;TEST -5.0V X G1
007132  104000                          PRINT
007134  014521                          MES24                   ;TEXT 'SWITCH VOLTAGE NEG.'
007136  104012                          WAITGN
007140  000000                          G1                      ;GAIN X1
007142  015252                          NEG500                  ;SHOULD=-5.0V
007144  015334                          GM50X1                  ;SAVE VALUE

                                        ;TEST +2.5V X G1
007146  104000                  GT2:    PRINT
007150  014536                          MES25                   ;TEXT '+2.5V'
007152  104012                          WAITGN
007154  000000                          G1                      ;GAIN X1
007156  015242                          POS250
007160  015266                          GP25X1                  ;SAVE VALUE
```

```
                                        ;TEST +2.5V X G2 (5.0V)
007162  104013                                  TAKEGN
007164  000010                                  G2                      ;GAIN X2
007166  015240                                  POS500                  ;SHOULD=+5.0V
007170  015276                                  GP25X2                  ;SAVE VALUE
007172  005767  005700                          TST     ADSIGN
007176  001414                                  BEQ     GT3             ;BRANCH IF UNIPOLAR
                                        ;TEST -2.5V X G1
007200  104000                                  PRINT
007202  014521                                  MES24                   ;TEXT SWITCH VOLTAGE NEG.
007204  104012                                  WAITGN
007206  000000                                  G1                      ;GAIN X1
007210  015254                                  NEG250
007212  015336                                  GM25X1                  ;SAVE VALUE

                                        ;TEST -2.5V X G2
007214  006267  005702                          ASR     KSTOR1          ;=TO -5.0V
007220  104013                                  TAKEGN
007222  000010                                  G2                      ;GAIN X2
007224  015252                                  NEG500
007226  015346                                  GM25X2                  ;SAVE VALUE

                                        ;TEST +1.25V X G1
007230  104000                          GT3:    PRINT
007232  014545                                  MES26                   ;TEXT '+1.25V'
007234  104012                                  WAITGN
007236  000000                                  G1                      ;GAIN X1
007240  015244                                  POS125
007242  015270                                  GP12X1                  ;SAVE VALUE

                                        ;TEST '+1.25V X G2
007244  104013                                  TAKEGN
007246  000010                                  G2                      ;GAIN X2
007250  015242                                  POS250                  ;=TO +2.5V
007252  015300                                  GP12X2                  ;SAVE VALUE
                                        ;TEST '+1.25 X G4
007254  104013                                  TAKEGN
007256  000020                                  G4                      ;GAIN X4
007260  015240                                  POS500
007262  015310                                  GP12X4                  ;SAVE VALUE
007264  005767  005606                          TST     ADSIGN
007270  001416                                  BEQ     GT4             ;BRANCH IF UNIPOLAR

                                        ;TEST '-1.25V X G1
007272  104000                                  PRINT
007274  014521                                  MES24                   ;TEXT 'SWITCH VOLTAGE NEG.'
007276  104012                                  WAITGN
007300  000000                                  G1                      ;GAIN X1
007302  015256                                  NEG125
007304  015340                                  GM12X1                  ;SAVE VALUE
```

```
                                        ;TEST -1.25V X G2
007306  104013                                  TAKEGN
007310  000010                                  G2                      ;GAIN X2
007312  015254                                  NEG250
007314  015350                                  GM12X2                  ;SAVE VALUE
                                        ;TEST -1.25V X G4
007316  104013                                  TAKEGN
007320  000020                                  G4                      ;GAIN X4
007322  015252                                  NEG500                  ;SHOULD = 5.0V
007324  015360                                  GM12X4                  ;SAVE VALUE

                                        ;TEST +0.625V X G1
007326  104000                          GT4:    PRINT
007330  014554                                  MES27                   ;TEXT '+0.625V'
007332  104012                                  WAITGN
007334  000000                                  G1
007336  015246                                  POS625                  ;SHOULD = +0.625V
007340  015272                                  GP62X1                  ;SAVE VALUE
                                        ;TEST +0.625V X G2
007342  104013                                  TAKEGN
007344  000010                                  G2                      ;GAIN X2
007346  015244                                  POS125                  ;SHOULD = +1.25V
007350  015302                                  GP62X2                  ;SAVE IT

                                        ;TEST +0.625V X G4
007352  104013                                  TAKEGN
007354  000020                                  G4                      ;GAIN X4
007356  015242                                  POS250                  ;SHOULD = +2.5V
007360  015312                                  GP62X4                  ;SAVE IT

                                        ;TEST +0.625V X G8
007362  104013                                  TAKEGN
007364  000030                                  G8                      ;GAIN X8
007366  015240                                  POS500                  ;SHOULD = +5.00V
007370  015322                                  GP62X8                  ;SAVE IT
007372  005767  005500                          TST     ADSIGN
007376  001422                                  BEQ     GT5             ;BRANCH IF UNIPOLAR

                                        ;TEST -0.625V X G1
007400  104000                                  PRINT
007402  014521                                  MES24                   ;SWITCH VOLTAGE NEG.
007404  104012                                  WAITGN
007406  000000                                  G1                      ;GAIN X1
007410  015260                                  NEG625                  ;SHOULD = -0.625V
007412  015342                                  GM62X1
                                        ;TEST -0.625V X G2
007414  104013                                  TAKEGN
007416  000010                                  G2                      ;GAIN X2
007420  015256                                  NEG125                  ;SHOULD = -1.25V
007422  015352                                  GM62X2                  ;SAVE IT
```

```
                                        ;TEST -0.625V X G4
007424  104013                                  TAKEGN
007426  000020                                  G4                      ;GAIN X4
007430  015254                                  NEG250                  ;SHOULD = -2.5V
007432  015362                                  GM62X4

                                        ;TEST -0.625V X G8
007434  104013                                  TAKEGN
007436  000030                                  G8                      ;GAIN X8
007440  015252                                  NEG500                  ;SHOULD = -5.00V
007442  015372                                  GM62X8

                                        ;TEST +0.3125V X G1
007444  104000                          GT5:    PRINT
007446  014564                                  MES28                   ;TEXT '+0.3125V'
007450  104012                                  WAITGN
007452  000000                                  G1                      ;GAIN X1
007454  015250                                  POS312                  ;SHOULD = +0.3125V
007456  015274                                  GP31X1                  ;SAVE IT

                                        ;TEST +0.3125V X G2
007460  104013                                  TAKEGN
007462  000010                                  G2                      ;GAIN X2
007464  015246                                  POS625                  ;SHOULD = +0.625V
007466  015304                                  GP31X2

                                        ;TEST +0.3125V X G4
007470  104013                                  TAKEGN
007472  000020                                  G4                      ;GAIN X4
007474  015244                                  POS125                  ;SHOULD = +1.25V
007476  015314                                  GP31X4

                                        ;TEST +0.3125V X G8
007500  104013                                  TAKEGN
007502  000030                                  G8                      ;GAIN X8
007504  015242                                  POS250                  ;SHOULD = +2.50V
007506  015324                                  GP31X8
007510  005767  005362                          TST     ADSIGN
007514  001422                                  BEQ     GT6             ;BRANCH IS UNIPOLAR

                                        ;TEST -0.3125V X G1
007516  104000                                  PRINT
007520  014521                                  MES24                   ;TEXT 'SWITCH NEG.'
007522  104012                                  WAITGN
007524  000000                                  G1                      ;GAIN X1
007526  015262                                  NEG312                  ;SHOULD = -0.3125V
007530  015344                                  GM31X1

                                        ;TEST -0.3125V X G2
007532  104013                                  TAKEGN
007534  000010                                  G2                      ;GAIN X2
007536  015260                                  NEG625                  ;SHOULD = -0.625V
007540  015354                                  GM31X2
```

```
                                        ;TEST -0.3125V X G4
007542  104013                                  TAKEGN
007544  000020                                  G4                      ;GAIN X4
007546  015256                                  NEG125                  ;SHOULD = -1.25V
007550  015364                                  GM31X4

                                        ;TEST -0.3125V X G8
007552  104013                                  TAKEGN
007554  000030                                  G8                      ;GAIN X8
007556  015254                                  NEG250                  ;SHOULD = -2.50V
007560  015374                                  GM31X8

                                        ;TEST +0.1563V X G2
007562  104000                          GT6:    PRINT
007564  014575                                  MES29                   ;TEXT '+0.1563V'
007566  104012                                  WAITGN
007570  000010                                  G2                      ;GAIN X2
007572  015250                                  POS312                  ;SHOULD = +0.3125V
007574  015306                                  GP15X2

                                        ;TEST +0.1563V X G4
007576  104013                                  TAKEGN
007600  000020                                  G4                      ;GAIN X4
007602  015246                                  POS625                  ;SHOULD = +0.625V
007604  015316                                  GP15X4

                                        ;TEST  0.1563V X G8
007606  104013                                  TAKEGN
007610  000030                                  G8                      ;GAIN X8
007612  015244                                  POS125                  ;SHOULD = +1.25V
007614  015326                                  GP15X8
007616  005767  005254                          TST     ADSIGN
007622  001416                                  BEQ     GT7             ;BRANCH IF UNIPOLAR

                                        ;TEST -0.1563V X G2
007624  104000                                  PRINT
007626  014521                                  MES24                   ;TEXT 'SWIT^H NEG.
007630  104012                                  WAITGN
007632  000010                                  G2                      ;GAIN 'X2'
007634  015262                                  NEG312                  ;SHOULD = -0.3125V
007636  015356                                  GM15X2

                                        ;TEST -0.1563V X G4
007640  104013                                  TAKEGN
007642  000020                                  G4                      ;GAIN X4
007644  015260                                  NEG625                  ;SHOULD = -0.625V
007646  015366                                  GM15X4

                                        ;TEST -0.1563V X G8
007650  104013                                  TAKEGN
007652  000030                                  G8                      ;GAIN X8
007654  015256                                  NEG125                  ;SHOULD = -1.25V
007656  015376                                  GM15X8
```

```
                                        ;TEST +0.0781V X G4
007660  104000                          GT7:    PRINT
007662  014606                                  MES30                   ;TEXT '+0.0781V'
007664  104012                                  WAITGN
007666  000020                                  G4                      ;GAIN X4
007670  015250                                  POS312                  ;SHOULD = +0.3125V
007672  015320                                  GP07X4

                                        ;TEST +0.0781V X 8
007674  104013                                  TAKEGN
007676  000030                                  G8                      ;GAIN X8
007700  015246                                  POS625                  ;SHOULD = +0.625V
007702  015330                                  GP07X8
007704  005767  005166                          TST     ADSIGN
007710  001412                                  BEQ     GT8

                                        ;TEST -0.0781V X G4
007712  104000                                  PRINT
007714  014521                                  MES24                   ;TEXT 'SWITCH NEG.'
007716  104012                                  WAITGN
007720  000020                                  G4                      ;GAIN X4
007722  015262                                  NEG312                  ;SHOULD = -0.3125V
007724  015370                                  GM07X4

                                        ;TEST -0.0781V X G8
007726  104013                                  TAKEGN
007730  000030                                  G8                      ;GAIN X8
007732  015260                                  NEG625                  ;SHOULD = -0.625V
007734  015400                                  GM07X8

                                        ;TEST +0.0390V X G8
007736  104000                          GT8:    PRINT
007740  014617                                  MES31                   ;TEXT '+0.0390V'
007742  104012                                  WAITGN
007744  000030                                  G8                      ;GAIN X8
007746  015250                                  POS312                  ;SHOULD = +0.3125V
007750  015332                                  GP03X8
007752  005767  005120                          TST     ADSIGN
007756  001406                                  BEQ     GT9

                                        ;TEST -0.0390V X G8
007760  104000                                  PRINT
007762  014521                                  MES24                   ;TEXT 'SWITCH NEG.'
007764  104012                                  WAITGN
007766  000030                                  G8
007770  015262                                  NEG312                  ;SHOULD = -0.3125V
007772  015402                                  GM03X8
```

```
                                        ;TYPE OUT A HISTOGRAM OF ALL THE GAIN AVERAGES.

007774  012767  015264  000142  GT9:    MOV     #GP5OX1,AVGTAB  ;SET UP GAIN TABLE
010002  012767  000002  005112          MOV     #2,KSTOR1
010010  104000                          PRINT
010012  014630                          MES32                   ;TYPE TABLE 'HEADER'.
010014  012767  000005  003372          MOV     #5,ICOUNT       ;SET UP PRINT ROUTINE
010022  104000                  GT10:   PRINT
010024  014765                          MES34                   ;TYPE GAIN X1 VALUES
010026  104014                          PRTAVG                  ;TYPE OUT AVERAGES X1
010030  104000                          PRINT
010032  014775                          MES35                   ;TYPE GAIN X2
010034  012767  000001  005056          MOV     #1,COUNT
010042  104015                          SIXDSH                  ;TYPE DASHES
010044  104014                          PRTAVG                  ;TYPE OUT AVERAGES X2
010046  104000                          PRINT
010050  015005                          MES36
010052  012767  000002  005040          MOV     #2,COUNT
010060  104015                          SIXDSH                  ;TYPE DASHES
010062  104014                          PRTAVG                  ;TYPE OUT AVERAGES X4
010064  104000                          PRINT
010066  015015                          MES37
010070  012767  000003  005022          MOV     #3,COUNT
010076  104015                          SIXDSH
010100  104014                          PRTAVG                  ;TYPE OUT AVERAGES X8
010102  005767  004770                  TST     ADSIGN
010106  001002                          BNE     GT11            ;IF UNIPOLAR, TYPE OUT NEG. COUNTS
010110  000167  176752                  JMP     GTO             ;OTHERWISE RESTART GAIN TEST
010114  005367  005002          GT11:   DEC     KSTOR1
010120  001002                          BNE     .+6
010122  000167  176740                  JMP     GTO
010126  104000                          PRINT
010130  014045                          CRLF
010132  000733                          BR      GT10
                                        ;SUBROUTINE TO TYPE OUT '5' AVERAGES FOR THE GAIN TEST HISTOGRAM.

010134  016767  003254  004774  XPRTAV: MOV     ICOUNT,TEMP3
010142  104010                  XPTA1:  PRTOCT                  ;PRINT OCTAL VALUE OF GAIN AVERAGE
010144  015264                  AVGTAB: GP5OX1
010146  062767  000002  177770          ADD     #2,AVGTAB       ;UPDATE GAIN TABLE
010154  012767  000002  000644          MOV     #2,SPACEX
010162  104007                          SPACE                   ;TYPE '2' SPACES
010164  005367  004746                  DEC     TEMP3
010170  001364                          BNE     XPTA1           ;IF NOT DONE, PRINT NEXT AVG.
010172  000002                          RTI
                                        ;SUBROUTINE TO TYPE OUT 'N' SETS OF SIX DASHES

010174  104000                  DASH6:  PRINT
010176  015025                          MES38
010200  005367  004714                  DEC     COUNT
010204  001373                          BNE     DASH6
010206  000002                          RTI
```

```
010210  104011                  XWATGN: TTYIN                          ;WAIT FOR 'CR' BEFORE CONTINUING
010212  117667  000000  004654          MOVB    @(SP),ADWRD2           ;GET SPECIFIED GAIN AND SET IT UP.
010220  062716  000002                  ADD     #2,(SP)                ;SET UP THE ADDRESS OF TRUE VOLTAGE
010224  017667  000000  000214          MOV     @(SP),PRTADR           ;SAVE ADDRESS OF TRUE VOLTAGE
010232  062716  000002                  ADD     #2,(SP)                ;SET UP STORAGE ADDRESS FOR VOLTAGE
010236  017667  000000  004660          MOV     @(SP),KSTOR2           ;SAVE ADDRESS
010244  062716  000002                  ADD     #2,(SP)                ;SET UP STACK TO EXIT
010250  104016                  GLOOP:  TSTTKS                         ;TEST FOR KEYBOARD FLAG
010252  012767  001000  004640          MOV     #1000,COUNT            ;SET UP TO TAKE '512' CONVERSIONS
010260  004767  001162                  JSR     PC,ADCNVT              ;TAKE THE CONVERSIONS
010264  104004                          CMPUTE                         ;COMPUTE THE AVERAGE
010266  016777  004670  004630          MOV     AVRAGE,@KSTOR2         ;SAVE THE AVERAGE VALUE
010274  104005                          CATORIZ                        ;CATAGORIZE THE COUNT SPREAD
010276  026777  004660  000142          CMP     AVRAGE,@PRTADR         ;IS AVERAGE =TO KNOWN VALUE?
010304  001414                          BEQ     GANEXT                 ;EXIT IF EQUAL
010306  026777  004652  000132          CMP     AVERP1,@PRTADR         ;IS AVERAGE =TO KNOWN VALUE +1?
010314  001410                          BEQ     GANEXT                 ;EXIT IF EQUAL
010316  027767  000124  004634          CMP     @PRTADR,AVERM1         ;IS AVERAGE =TO KNOWN VALUE -1?
010324  001404                          BEQ     GANEXT                 ;EXIT IF EQUAL
010326  032777  040000  004524  GAINER: BIT     #SW14,@SWR             ;TEST FOR INHIBIT SCOPE LOOPING
010334  001401                          BEQ     GERR1                  ;BRANCH IF NOT SET
010336  000002                  GANEXT: RTI
010340  032777  020000  004512  GERR1:  BIT     #SW13,@SWR             ;TEST FOR PRINT INHIBIT
010346  001340                          BNE     GLOOP                  ;BRANCH IS SW SET
010350  005767  004564                  TST     MESPRT                 ;TEST IF TITLE HAS BEEN TYPED
010354  001005                          BNE     GERR2                  ;BRANCH IF YES
010356  052767  000001  004554          BIS     #1,MESPRT              ;OTHERWISE TYPE ERROR HEADER
010364  104000                          PRINT
010366  014735                          MES33                          ;TEXT 'GAIN
010370  105767  004500          GERR2:  TSTB    ADWRD2                 ;TEST FOR 'G1'
010374  001003                          BNE     GERR3                  ;BRANCH IF NOT
010376  104000                          PRINT
010400  014765                          MES34                          ;TEXT '1'
010402  000420                          BR      GERR6
010404  122767  000010  004462  GERR3:  CMPB    #10,ADWRD2             ;TEST FOR GX2
010412  001003                          BNE     GERR4                  ;BRANCH IF NOT -
010414  104000                          PRINT
010416  014775                          MES35                          ;TEXT '2'
010420  000411                          BR      GERR6
010422  122767  000020  004444  GERR4:  CMPB    #20,ADWRD2             ;TEST FOR GAIN OF '4'
010430  001003                          BNE     GERR5                  ;BRANCH AND PRINT GX8
010432  104000                          PRINT
010434  015005                          MES36                          ;TEXT '4'
010436  000402                          BR      GERR6
010440  104000                  GERR5:  PRINT
010442  015015                          MES37                          ;TEXT '8'
010444  104010                  GERR6:  PRTOCT
010446  000000                  PRTADR: 0                              ;TYPE VOLTAGE VALUE
010450  104007                          SPACE                          ;TYPE SPACE
010452  104007                          SPACE
010454  104010                          PRTOCT
010456  015162                          AVRAGE                         ;TYPE AVERAGE
010460  104000                          PRINT
010462  014045                          CRLF
010464  000671                          BR      GLOOP                  ;RETEST GAIN AVERAGE
                                ;*******************************************************************
```

```
                                        ;RECOVERY TEST
                                        ;*******************************************************************

                                        ;THIS TEST IS DESIGNED TO TEST RECOVERY OF THE A/D CONVERTER VIA ACCEPT-
                                        ;ING TWO (2) CHANNEL AND GAIN INPUTS FROM THE TELETYPE AND THEN TAKE A
                                        ;SERIES OF EIGHT CONVERSIONS ON EACH CHANNEL AND TYPING OUT THE CON-
                                        ;VERSION VALUES OF THE 2ND CHANNEL IN THE ORDER THEY WERE TAKEN.

010466  012767  010506  004414  RECVRY: MOV     #RECVY1,AVECTR   ;SET UP THE '↑A' RETURN ADDRESS
010474  104000                          PRINT
010476  014150                          MES8                     ;TEXT 'RECOVERY TEST'
010500  012767  000010  002706          MOV     #10,ICOUNT       ;SET UP TO PRINT '8' VALUES
010506  104000                  RECVY1: PRINT                    ;REQUEST CHANNELS
010510  014276                          MES14
010512  104011                          TTYIN                    ;WAIT FOR INPUT
010514  104001                          DECOCT                   ;CONVERT TO OCTAL
010516  016767  000714  004376          MOV     BCDTAB,KSTOR1    ;SAVE 1ST CH.
010524  016767  000710  004372          MOV     BCDTAB+2,KSTOR2  ;SAVE 2ND CH.
010532  104003                          GAININ                   ;GET THE 1ST GAIN SETTING
010534  116767  004334  004364          MOVB    ADWRD2,KSTOR3    ;SAVE IT
010542  016767  000672  000666          MOV     BCDTAB+2,BCDTAB  ;SET UP TO CONVERT 2ND GAIN
010550  104017                          GAINXN                   ;DECODE 2ND GAIN
010552  116767  004316  004350          MOVB    ADWRD2,KSTOR4    ;SAVE IT
010560  104016                  RECVY2: TSTTKS                   ;CHECK FOR KEYBOARD FLAG
010562  012767  000010  004330          MOV     #10,COUNT        ;SETUP TO TAKE '8' CONVERSIONS ON 1ST CH.
010570  116767  004326  004277          MOVB    KSTOR1,ADWRD2+1  ;LOAD 1ST CH.
010576  116767  004324  004270          MOVB    KSTOR3,ADWRD2    ;SET UP GAIN
010604  004767  000636                  JSR     PC,ADCNVT        ;TAKE THE CONVERSIONS
010610  116767  004310  004257          MOVB    KSTOR2,ADWRD2+1  ;SET UP 2ND CH.
010616  116767  004306  004250          MOVB    KSTOR4,ADWRD2    ;LOAD 2ND GAIN
010624  004767  000616                  JSR     PC,ADCNVT        ;TAKE 2ND SERIES OF CONVERSIONS
010630  032777  020000  004222          BIT     #SW13,@SWR       ;TEST THE PRINT INHIBIT SW
010636  001350                          BNE     RECVY2           ;BRANCH IF SET
010640  104000                          PRINT
010642  014171                          MES9                     ;TEXT 'CH.'
010644  016702  004254                  MOV     KSTOR2,R2
010650  104006                          BINDEC                   ;TYPE 2ND CH.
010652  104007                          SPACE
010654  012767  015400  177262          MOV     #ADBUFF,AVGTAB
010662  104014                          PRTAVG                   ;PRINT VALUES OF 2ND CH.
010664  000735                          BR RECVY2                ;DO IT AGAIN
```

```
                                        ;ROUTINE TO LOOP THUR A SINGLE LOGIC SUBTEST.  ENTERED FROM THE 'MONITOR'
                                        ;VIA TYPING 'TNN' WHERE 'NN' IS EQUATED TO THE 'PC' OF A SUBTEST.
                                        ;NOTE THAT 'SW11' MUST BE '0' (DOWN) TO RUN THIS TEST.

010666  004767  170600          TESTX:  JSR     PC,SETUP        ;MAIN INITIALIZATION FOR LOGIC TEST
010672  012701  011272                  MOV     #INBUF,R1
010676  005067  004220                  CLR     KSTOR1
010702  042711  177770          TSTA:   BIC     #177770,(R1)    ;STRIPE NO. TO OCTAL
010706  062167  004210                  ADD     (R1)+,KSTOR1    ;ADD TO LAST RESULT
010712  005367  004200                  DEC     CHRCNT
010716  001407                          BEQ     TSTXB
010720  006367  004176                  ASL     KSTOR1
010724  006367  004172                  ASL     KSTOR1
010730  006367  004166                  ASL     KSTOR1
010734  000762                          BR      TSTA

010736  022767  005754  004156  TSTXB:  CMP     #TSTDN,KSTOR1   ;IS NO. WITHIN LIMITS OF THE LOGIC TEST
010744  100002                          BPL     .+6             ;CONTINUE IF YES
010746  000167  170512                  JMP     INIT3           ;OTHERWISE RETURN TO MONITOR
010752  062767  000002  004142          ADD     #2,KSTOR1       ;ADD '2' TO POINT TO INSTRUCTION AFTER SCOPE
010760  005067  002432          XLOOP:  CLR     SCOPEF          ;KEEP COUNT AT ZERO
010764  012767  010760  002426          MOV     #XLOOP,RETURN   ;LOAD SCOPE LOOP RETURN POINTER
010772  000177  004124                  JMP     @KSTOR1         ;JUMP TO TEST

                                        ;SUBROUTINE TO ISSUE N SPACES
                                        ;N IS ONE PLUS VALUE CONTAINED IN SPACEX
                                        ;SPACEX IS CLEARED WITHIN THE SUBROUTINE, SO THAT A CALL ON
                                        ;SPACE WITHOUT LOADING SPACEX ISSUES ONLY ONE SPACE
010776  105777  004052          XSPACE: TSTB    @TPS            ;WAIT FOR TTY READY
011002  100375                          BPL     .-4
011004  012777  000240  004044          MOV     #240,@TPB       ;OUTPUT A SPACE
011012  005367  000010                  DEC     SPACEX          ;DECREMENT COUNT
011016  003367                          BGT     XSPACE          ;LOOP IF NOT DONE
011020  005067  000002                  CLR     SPACEX          ;RESET COUNT TO ZERO
011024  000002                          RTI                     ;RETURN
011026  000000                  SPACEX: 0
```

```
                                        ;KEYBOARD SERVICE ROUTINE

011030  012704  011272          XTTYIN: MOV     #INBUF,R4       ;SETUP CHARACTER BUFFER
011034  005067  004056          NEWIN:  CLR     CHRCNT          ;CLEAR CHARACTER COUNTER
011040  105777  004004          INPUTA: TSTB    @TKS            ;CHARACTER READY?
011044  100375                          BPL     INPUTA          ;NO, WAIT IT OUT
011046  017701  004000                  MOV     @TKB,R1         ;SAVE CHARACTER
011052  042701  000200                  BIC     #200,R1         ;STRIPE PARITY BIT
011056  120127  000060                  CMPB    R1,#60          ;IS IT A SPECIAL CHARACTER
011062  100427                          BMI     SPCHR           ;YES, TEST IT
011064  122701  000137                  CMPB    #137,R1
011070  100424                          BMI     SPCHR
011072  122701  000124                  CMPB    #124,R1         ;IS CHAR 'T'
011076  001004                          BNE     INPUTB          ;NO.
011100  012767  000372  003772          MOV     #372,SUBX       ;YES,SET THE SUBTEST CALL SW.
011106  000407                          BR      OUTPTA          ;IGNORE'T' AND SAVE ADDRESS
011110  010124                  INPUTB: MOV     R1,(R4)+        ;SAVE CHARACTER
011112  005267  004000                  INC     CHRCNT          ;INCREMENT THE CHARACTER COUNT.
011116  022767  000011  003772          CMP     #11,CHRCNT
011124  100457                          BMI     SPCHR5          ;TYPE '?' IF TOO MANT CHAR.
011126  105777  003722          OUTPTA: TSTB    @TPS            ;ECHO CHARACTER
011132  100375                          BPL     OUTPTA
011134  110177  003716                  MOVB    R1,@TPB
011140  000737                          BR      INPUTA          ;WAIT FOR NEXT CHARACTER
                                        ;SUBROUTINE TO TEST FOR SPECIAL CHARACTERS :'↑A','↑C','+','CR',',' OR 'RUBOUT'

011142  122701  000001          SPCHR:  CMPB    #1,R1           ;CHAR.='↑A'
011146  001005                          BNE     SPCHR1          ;NO, NOT '↑A'
011150  104000                          PRINT                   ;ECHO '↑A'
011152  014042                          CNTRLA                  ;RESTORE 'SP'
011154  022626                          POP2SP
011156  000177  003726                  JMP     @AVECTR         ;YES, EXIT VIA '↑A' VECTOR ADDRESS.
011162  122701  000003          SPCHR1: CMPB    #3,R1           ;CHAR.='↑C'
011166  001002                          BNE     .+6             ;NO,NOT '↑C'
011170  000167  170074                  JMP     MONITR          ;YES, EXIT TO MONITOR
011174  122701  000177                  CMPB    #177,R1         ;CHAR. = 'RUBOUT'
011200  001011                          BNE     SPCHR2          ;IGNORE CHAR. & EXIT
011202  005767  003710                  TST     CHRCNT          ;IS RUBOUT LEGAL?
011206  001714                          BEQ     INPUTA          ;NO, IGNORE IT
011210  005367  003702                  DEC     CHRCNT
011214  012701  000134                  MOV     #134,R1         ;TYPE '\' TO INDICATE RUBOUUT
011220  005744                          TST     -(R4)           ;POP OFF LAST CHARACTER
011222  000741                          BR      OUTPTA          ;WAIT FOR NEXT CHARACTER
011224  122701  000053          SPCHR2: CMPB    #53,R1          ;TEST FOR '+'
011230  001004                          BNE     SPCHR3          ;BRANCH IF NO
011232  012767  000177  003636          MOV     #177,ADSIGN     ;YES, IDICATES UNIPOLAR
011240  000732                          BR      OUTPTA          ;WAIT NEXT CHAR.
011242  122701  000054          SPCHR3: CMPB    #54,R1          ;TEST FOR ','
011246  001720                          BEQ     INPUTB          ;LEGAL CHAR., SAVE IT
011250  122701  000015          SPCHR4: CMPB    #15,R1          ;=TO 'CARRIAGE RETURN' TO TERMINATE?
011254  001003                          BNE     SPCHR5          ;NO, CONTINUE
011256  104000                          PRINT                   ;YES, TYPE 'CR-LF'
011260  014045                          CRLF
011262  000002                  EXTTY:  RTI                     ;EXIT
```

```
011264  104000                  SPCHRS: PRINT                   ;OTHERWISE TYPE '?'
011266  014052                          QMARK
011270  000661                          BR NEWIN                ;WAIT FOR NEW ENTRY
011272  000000                  INBUF:  0                       ;CHARACTER STORAGE BUFFER
        011306                          .=.+12
                                ;SUBROUTINE WILL CONVERT 'N' BCD WORDS (SEPARATED VIA COMMA'S)
                                ;WHICH WERE STORED IN A TABLE VIA 'TTYIN' TO OCTAL AND STORE THEM.

011306  012704  011272          BCDBIN: MOV     #INBUF,R4       ;SETUP ASCII STORAGE TABLE
011312  012703  011436                  MOV     #BCDTAB,R3      ;TABLE FOR STORAGE OF CONVERTED WORDS
011316  005067  000116                  CLR     BCDTAB+2
011322  005001                  BCDBN1: CLR     R1              ;REG. TO STORE RUNNING TOTAL
011324  005002                          CLR     R2              ;TEMP. STORAGE FOR 'R1'
011326  005767  003564          BCDBN2: TST     CHRCNT          ;END OF DATA?
011332  003426                          BLE     BCDEND          ;YES, EXIT
011334  005367  003556                  DEC     CHRCNT          ;DECREMENT CHARACTER COUNTER
011340  122714  000054                  CMPB    #54,(R4)        ;IS CHARACTER = TO ' ,'?
011344  001421                          BEQ     BCDEND          ;YES, DECODE NEW WORD
011346  121427  000060                  CMPB    (R4),#60
011352  002424                          BLT     BCDERR          ;TEST FOR LEGAL NO.
011354  021427  000071                  CMP     (R4),#71
011360  003021                          BGT     BCDERR
011362  042714  177760                  BIC     #177760,(R4)    ;STRIPE NO. TO BCD
011366  012400                          MOV     (R4)+,R0        ;SAVE NO. IN R0.
011370  010102                          MOV     R1,R2           ;SAVE CURRENT TOTAL
011372  006301                          ASL     R1              ;NX2
011374  006301                          ASL     R1              ;NX4
011376  006301                          ASL     R1              ;NX8
011400  060201                          ADD     R2,R1           ;NX9
011402  060201                          ADD     R2,R1           ;NX10
011404  060001                          ADD     R0,R1           ;N+NEW NO.
011406  000747                          BR      BCDBN2
011410  005724                  BCDEND: TST     (R4)+           ;UPDATE BUFFER
011412  010123                          MOV     R1,(R3)+        ;SAVE CONVERTED VALUE & SETUP TO SAVE NEXT
011414  005767  003476                  TST     CHRCNT          ;FINISHED?
011420  001340                          BNE     BCDBN1          ;NO, CONVERT NEXT WORD
011422  000002                          RTI                     ;YES, EXIT
011424  104000                  BCDERR: PRINT
011426  014052                          QMARK                   ;TYPE '?'
011430  104011                          TTYIN                   ;TO BE TYPED ON QUESTIONABLE ENTRIES.
011432  000167  177650                  JMP     BCDBIN
011436  000000                  BCDTAB: 0                       ;OCTAL STORAGE TABLE
011440  000000                          0
011442  000000                          0
011444  000000                          0
```

```
                                        ;SUBROUTINE TO TAKE 'N' CONVERSIONS AND STORE THEM IN AN A/D BUFFER, ROUTINE
                                        ;IS ENTERED WITH 'N' IN COUNT AND THE CH & GAIN TO BE CONVERTED IN 'ADWORD'.

011446  016767  003446  003456  ADCNVT: MOV     COUNT,TEMP1      ;SET UP NO. OF CONVERSIONS TO BE TAKEN
011454  012704  015400                  MOV     #ADBUFF,R4       ;SET UP BUFFER ADDRESS.
011460  005777  003404                  TST     @ADDBR           ;CLR A/D DONE
011464  116777  003404  003372          MOVB    ADWRD2,@ADCS     ;SET UP GAIN AND EXT CONTROL IF SET
011472  116777  003377  003366  CNVT1:  MOVB    ADWRD2+1,@ADCS0  ;LOAD CH. & START CONVERT IF NOT 'EXT'
011500  105777  003360                  TSTB    @ADCS            ;TEST A/D DONE
011504  100375                          BPL     .-4              ;WAIT FOR DONE
011506  005777  003352                  TST     @ADCS            ;TEST ERROR BIT
011512  100002                          BPL     CNVT2            ;BRANCH IT NOT SET
011514  104000                          PRINT                    ;OTHERWISE TYPE ERROR
011516  014230                          MES12                    ;TEXT 'ERROR BIT SET'.
011520  017724  003344          CNVT2:  MOV     @ADDBR,(R4)+     ;SAVE DATA
011524  005367  003402                  DEC     TEMP1            ;DECREMENT COUNTER
011530  003360                          BGT     CNVT1            ;IF NOT '0' TAKE NEXT CO' 'ERSION
011532  000207                          RTS     PC

                                        ;POWER FAIL HANDLER
011534  010046                  PWRFAL: MOV     R0,-(SP)
011536  010146                          MOV     R1,-(SP)
011540  010246                          MOV     R2,-(SP)
011542  010346                          MOV     R3,-(SP)
011544  010446                          MOV     R4,-(SP)
011546  010546                          MOV     R5,-(SP)
011550  016746  166250                  MOV     24,-(SP)
011554  010667  003334                  MOV     SP,PROC
011560  012767  011570  166236          MOV     #PWRUP,24
011566  000000                          HALT
                                        ;POWER UP HANDLER
011570  012777  000340  003250  PWRUP:  MOV     #340,@PSW
011576  016706  003312                  MOV     PROC,SP
011602  012667  166216                  MOV     (SP)+,24
011606  012605                          MOV     (SP)+,R5
011610  012604                          MOV     (SP)+,R4
011612  012603                          MOV     (SP)+,R3
011614  012602                          MOV     (SP)+,R2
011616  012601                          MOV     (SP)+,R1
011620  012600                          MOV     (SP)+,R0
011622  000005                          RESET
011624  104000                          PRINT
011626  014462                          MES21
011630  000167  167434                  JMP     MONITR
```

```
                                        ;SUBROUTINE TO INPUT A 'GAIN FROM THE TELETYPE AND STORE IT IN THE 'ADWORD'

011634  104000                  XGAINA: PRINT
011636  014330                          MES18                   ;TEXT 'GAIN?'
011640  104011                          TTYIN                   ;WAIT FOR INPUT
011642  104001                          DECOCT                  ;CONVERT TO OCTAL
011644  022767  000001  177564  XGAINB: CMP     #1,BCDTAB        ;TEST FOR '1'
011652  001003                          BNE     XGAIN2          ;IF NOT '1' TEST FOR '2'
011654  105067  003214                  CLRB    ADWRD2          ;OTHERWISE SELECT GAIN X1
011660  000002                          RTI
011662  022767  000002  177546  XGAIN2: CMP     #2,BCDTAB
011670  001004                          BNE     XGAIN4
011672  112767  000010  003174          MOVB    #10,ADWRD2
011700  000002                          RTI
011702  022767  000004  177526  XGAIN4: CMP     #4,BCDTAB
011710  001004                          BNE     XGAIN8
011712  112767  000020  003154          MOVB    #20,ADWRD2
011720  000002                          RTI
011722  022767  000010  177506  XGAIN8: CMP     #10,BCDTAB
011730  001341                          BNE     XGAINA          ;ILLEGAL ENTRY, TRY AGAIN
011732  112767  000030  003134          MOVB    #30,ADWRD2
011740  000002                          RTI
```

```
                                        ;EMT DISPATCH SERVICE ROUTINE
                                        ;ARGUMENT OF EMT IS EXTRACTED AND USED AS OFFSET TO OBTAIN POINTER
                                        ;TO THE SELECTED SUBROUTINE.

011742  011646                          EMTSRV: MOV     (SP),-(SP)      ;GET PC FOR TO RETURN
011744  162716  000002                          SUB     #2,(SP)         ;PC OF EMT
011750  017616  000000                          MOV     @(SP),(SP)      ;GET EMT
011754  005716                                  TST     (SP)            ;IS EMT VALID?
011756  001001                                  BNE     EMTOK
011760  000000                                  HALT                    ;INVALID EMT
011762  006316                          EMTOK:  ASL     (SP)            ;MULTIPLY EMT ARG BY '2'
011764  042716  177001                          BIC     #177001,(SP)    ;CLEAR UNWANTED BITS
011770  062716  012002                          ADD     #EMTTAB,(SP)    ;POINTER TO SUBROUTINE ADDRESS
011774  017616  000000                          MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
012000  000136                                  JMP     @(SP)+          ;GO TO SUBROUTINE

                                        ;EMT DISPATCH TABLE

012002  012042                          EMTTAB: TYPMES                  ;MESSAGE PRINT ROUTINE
012004  011306                                  BCDBIN                  ;DECIMAL TO BINARY CONVERSION ROUTINE
012006  013332                                  SCOPEC                  ;LOGIC TEST SCOPE ROUTINE
012010  011634                                  XGAINA                  ;REQUEST A 'GAIN' FROM THE TTY.
012012  012334                                  CMPTE                   ;SUBROUTINE TO COMPUTE THE AVG
012014  012526                                  CATORZ                  ;SUBROUTINE TO COMPUTE 'COUNT SPREAD'
012016  012136                                  DECPRT                  ;SUBROUTINE TO CONVERT OCT TO DEC + PRINT
012020  010776                                  XSPACE                  ;SUBROUTINE TO TYPE SPACES
012022  013112                                  OCTPRT                  ;OCTAL PRINT ROUTINE
012024  011030                                  XTTYIN                  ;TELEPRINTER SERVICE ROUTINE
012026  010210                                  XWATGN                  ;GAIN TEST CONVERSION ROUTINE
012030  010212                                  XWATGN+2                ;GAIN TEST CONVERSION ROUTINE
012032  010134                                  XPRTAV                  ;SUBROUTINE TO PRINT OUT THE GAIN AVERAGES
012034  010174                                  DASH6                   ;SUBROUTINE TO TYPE OUT '6' DASHES
012036  013230                                  TKSFLG                  ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
012040  011644                                  XGAINB                  ;SUBROUTINE TO DECODE A GAIN FOR TTY
```

```
                                        ;MESSAGE PRINT ROUTINE, ENTERED VIA EMT DISPATCH HANDLER.
                                        ;ROUTINE PICKS UP CONTENTS OF THE 'PC' AND USES THIS AS
                                        ;THE ADDRESS OF MESSAGE TO BE TYPED.

012042  005077  003000                  TYPMES: CLR     @PSW            ;ENABLE KEYBOARD INTR.
012046  017605  000000                          MOV     @(SP),R5        ;GET THE MESSAGE ADDRESS FROM START
012052  062716  000002                          ADD     #2,(SP)         ;SET UP STACK TO EXIT
012056  105777  002772                  TYPERA: TSTB    @TPS
012062  100375                                  BPL     TYPERA          ;WAIT FOR TTY DONE
012064  122715  000100                          CMPB    #100,(R5)       ;TEST FOR '@'
012070  001001                                  BNE     TYPER1          ;BRANCH IF NO EQUAL
012072  000002                                  RTI                     ;OTHERWISE EXIT
012074  122715  000045                  TYPER1: CMPB    #45,(R5)        ;TEST FOR '%'
012100  001403                                  BEQ     TYPECL          ;IF = TYPE 'CR-LF'
012102  112577  002750                  TYPER2: MOVB    (R5)+,@TPB      ;OUTPUT CHAR.
012106  000763                                  BR      TYPERA
012110  012777  000015  002740          TYPECL: MOV     #15,@TPB        ;TYPE 'CR'
012116  105777  002732                          TSTB    @TPS
012122  100375                                  BPL     .-4
012124  012777  000012  002724                  MOV     #12,@TPB
012132  105725                                  TSTB    (R5)+           ;INCREMENT BUFFER
012134  000750                                  BR      TYPERA

                                        ;PRINT DECIMAL VALUE IN R2

012136  005077  002704                  DECPRT: CLR     @PSW            ;ENABLE TTY INTERRUPTS
012142  012767  177774  000146                  MOV     #-4,DIGCNT
012150  012767  012324  000144                  MOV     #DECPNT+2,DECPNT
012156  012767  000240  000134                  MOV     #240,ZERO
012164  012767  177777  000122          TYPT1:  MOV     #-1,DIGIT
012172  005267  000116                  TYPT2:  INC     DIGIT
012176  167702  000120                          SUB     @DECPNT,%0
012202  100373                                  BPL     TYPT2
012204  067702  000112                          ADD     @DECPNT,%2
012210  004767  000020                          JSR     PC,DECOUT
012214  005267  000076                          INC     DIGCNT
012220  001001                                  BNE     TYPT3
012222  000002                                  RTI
012224  062767  000002  000070          TYPT3:  ADD     #2,DECPNT
012232  000754                                  BR      TYPT1
012234  005767  000054                  DECOUT: TST     DIGIT
012240  001010                                  BNE     DEC1
012242  022767  177777  000046                  CMP     #-1,DIGCNT
012250  001404                                  BEQ     DEC1
012252  016767  000042  000034                  MOV     ZERO,DIGIT
012260  000406                                  BR      DEC2
012262  012767  000260  000030          DEC1:   MOV     #260,ZERO
012270  052767  000260  000016                  BIS     #260,DIGIT
012276  105777  002552                  DEC2:   TSTB    @TPS
012302  100375                                  BPL     .-4
012304  016777  000004  002544                  MOV     DIGIT,@TPB
012312  000207                                  RTS     7
```

```
012314  000000                          DIGIT:  0
012316  000000                          DIGCNT: 0
012320  000240                          ZERO:   240
012322  012324                          DECPNT: .+2
012324  001750                                  1000.
012326  000144                                  100.
012330  000012                                  10.
012332  000001                                  1.

                                        ;COMPUTE THE RESULTS OF 512 CONVERSIONS AS HIGH, LOW AND AVERAGE

012334  012767  000777  002570          CMPTE:  MOV     #777,TEMP1          ;SET UP TO COMPARE '511' NUMBERS
012342  005067  002574                          CLR     HIORDV              ;CLR HI ORDER DIVIDEND
012346  012704  015400                          MOV     #ADBUFF,R4          ;SET UP DATA BUFFER ADDRESS
012352  012467  002604                          MOV     (R4)+,AVRAGE        ;STORE 1ST VALUE AS AVERAGE
012356  016767  002600  002560                  MOV     AVRAGE,HIGH         ;HIGH
012364  016767  002572  002554                  MOV     AVRAGE,LOW          ;& LOW
012372  066767  002514  002562                  ADD     ADSIZE,AVRAGE       ;ADD OFFSET TO AVERAGE
012400  012467  002530                  GETDAT: MOV     (R4)+,TEMP2
012404  026767  002524  002532                  CMP     TEMP2,HIGH          ;IS NEW NO. GREATER THAN OLD NO.
012412  003403                                  BLE     TSLO                ;BRANCH IF NOT GREATER
012414  016767  002514  002522                  MOV     TEMP2,HIGH          ;OTHERWISE SAVE AS NEW HIGH
012422  026767  002506  002516          TSLO:   CMP     TEMP2,LOW
012430  003003                                  BGT     TAGA
012432  016767  002476  002506                  MOV     TEMP2,LOW           ;OTHERWISE SAVE AS NEW LOW
012440  066767  002446  002466          TAGA:   ADD     ADSIZE,TEMP2        ;ADD OFFSET TO MAKE ALL NO. POS.
012446  066767  002462  002506                  ADD     TEMP2,AVRAGE        ;ADD LOW ORDER
012454  005367  002462                          ADC     HIORDV              ;ADD CARRY TO HI ORDER
012460  005367  002446                          DEC     TEMP1
012464  001345                                  BNE     GETDAT              ;512 ADDITIONS?
012466  012767  000011  002436                  MOV     #11,TEMP1           ;YES, DIVIDE/512
012474  006267  002442                  AVGDAT: ASR     HIORDV
012500  006067  002456                          ROR     AVRAGE              ;SHIFT CARRY BIT INTO LO ORDER
012504  005367  002422                          DEC     TEMP1
012510  001371                                  BNE     AVGDAT              ;DONE?
012512  005567  002444                          ADC     AVRAGE              ;YES, ADD REMAINDER TO LO ORDER
012516  166767  002370  002436                  SUB     ADSIZE,AVRAGE       ;SUBTRACT OFFSET TO OBTAIN REAL AVERAGE
012524  000002                                  RTI
```

```
                                        ;SUBROUTINE TO CALCULATE THE PLUS & MINUS 5 COUNT LIMITS FROM AN AVERAGE

012526  012767  000005  002376  CATORZ: MOV     #5,TEMP1
012534  016767  002422  002372          MOV     AVRAGE,TEMP2     ;MOV AVER. TO WORK AREA
012542  066767  002344  002364          ADD     ADSIZE,TEMP2     ;MAKE AVG. POS.
012550  012703  015164                  MOV     #AVERP1,R3       ;SETUP DISTRIBUTION TABLE (POS.)
012554  005267  002354          FILE1:  INC     TEMP2            ;A=A+1
012560  016713  002350                  MOV     TEMP2,(R3)       ;SAVE A+1
012564  166723  002322                  SUB     ADSIZE,(R3)+     ;RESTORE ORIGINAL VALUE
012570  005367  002336                  DEC     TEMP1            ;SAVED '5' COUNTS?
012574  001367                          BNE     FILE1            ;BRANCH IF NO
                                        ;SET UP TABLE OF AVG. -1 TO -5
012576  012767  000005  002326          MOV     #5,TEMP1
012604  016767  002352  002322          MOV     AVRAGE,TEMP2     ;MOV AVG. TO WORK AREA.
012612  066767  002274  002314          ADD     ADSIZE,TEMP2
012620  012703  015162                  MOV     #AVRAGE,R3       ;SET UP DISTRIBUTION TABLE NEG.
012624  005367  002304          FILE2:  DEC     TEMP2            ;A=1-1
012630  016743  002300                  MOV     TEMP2,-(R3)      ;SAVE 'A-1'
012634  166713  002252                  SUB     ADSIZE,(R3)      ;RESTORE ORIGINAL NO. -1
012640  005367  002266                  DEC     TEMP1            ;SAVED '5' COUNTS?
012644  001367                          BNE     FILE2            ;BRANCH IF NO

                                        ;CATEGORIZE THE COUNT SPREAD AS '+6 & -6' COUNTS FROM THE AVERAGE

012646  012703  015176                  MOV     #ORLOW,R3        ;CLEAR COUNTS
012652  005023                  CATR1:  CLR     (R3)+
012654  022703  015230                  CMP     #ORHIGH+2,R3     ;FINISHED?
012660  001374                          BNE     CATR1            ;NO, CLEAR NEXT COUNTER
012662  012767  001001  002242          MOV     #1001,TEMP1      ;COMPARE '512' COUNTS
012670  012700  015400                  MOV     #ADBUFF,R0       ;SET UP A/D BUFFER
012674  005367  002232          CATR2:  DEC     TEMP1
012700  001437                          BEQ     CATR5            ;EXIT IF '0'
012702  012067  002226                  MOV     (R0)+,TEMP2
012706  026767  002262  002220          CMP     AVERP5,TEMP2
012714  100423                          BMI     OVRHI
012716  026767  002212  002224          CMP     TEMP2,AVERM5
012724  100422                          BMI     OVRLO
012726  005001                          CLR     R1
012730  012702  015150                  MOV     #AVERM5,R2
012734  022267  002174          CATR3:  CMP     (R2)+,TEMP2
012740  001405                          BEQ     CATR4
012742  005201                          INC     R1
012744  022701  000013                  CMP     #13,R1
012750  001371                          BNE     CATR3
012752  000000                          HALT                     ;FATAL ERROR MR. BOUSE!
012754  006301                  CATR4:  ASL     R1               ;MULTIPLY 'OFFSET' X2
012756  005261  015200                  INC     MINUS5(R1)
012762  000744                          BR      CATR2
012764  005267  002236          OVRHI:  INC     ORHIGH
012770  000741                          BR      CATR2
012772  005267  002200          OVRLO:  INC     ORLOW
012776  000736                          BR      CATR2
```

```
                                        ;ADD THE COUNTS AND SAVE TOTAL IN SPREADS OF '1-4'

013000  016767  002206  002222  CATR5:  MOV     AVGCNT,XSPRD1
013006  066767  002202  002214          ADD     PLUS1,XSPRD1
013014  066767  002170  002206          ADD     MINUS1,XSPRD1   ;=TO NO. COUNTS AT SPREAD OF '1'
013022  016767  002202  002202          MOV     XSPRD1,XSPRD2
013030  066767  002162  002174          ADD     PLUS2,XSPRD2
013036  066767  002144  002166          ADD     MINUS2,XSPRD2   ;=TO NO. COUNTS AT SPREAD OF  2'
013044  016767  002162  002162          MOV     XSPRD2,XSPRD3
013052  066767  002142  002154          ADD     PLUS3,XSPRD3
013060  066767  002120  002146          ADD     MINUS3,XSPRD3   ;=TO NO. COUNTS AT SPREAD OF '3'
013066  016767  002142  002142          MOV     XSPRD3,XSPRD4
013074  066767  002122  002134          ADD     PLUS4,XSPRD4
013102  066767  002074  002126          ADD     MINUS4,XSPRD4   ;=TO NO. COUNTS AT SPREAD OF '4'
013110  000002                          RTI                     ;EXIT

                                        ;SUBROUTINE TO TYPEOUT A '6' DIGIT OCTAL NO. THE 'PC' CONTAINS
                                        ;THE ADDRESS OF 'WORD' TO BE TYPED

013112  005077  001730          OCTPRT: CLR     @PSW            ;ENABLE KEYBOARD INTR.
013116  017605  000000                  MOV     @(SP),R5        ;THE ADDRESS OF WORD TO BE TYPED
013122  062716  000002                  ADD     #2,(SP)         ;SET UP STACK TO EXIT
013126  012767  000006  001776          MOV     #6,TEMP1
013134  012767  000376  000064          MOV     #376,MASK       ;MASK FOR FIRST BIT
013142  000401                          BR      .+4
013144  006115                  SHIFT:  ROL     (R5)
013146  006115                          ROL     (R5)
013150  006115                          ROL     (R5)
013152  111567  001756                  MOVB    (R5),TEMP2
013156  146767  000044  001750          BICB    MASK,TEMP2
013164  052767  000260  001742          BIS     #260,TEMP2
013172  132777  000200  001654          BITB    #200,@TPS
013200  100374                          BPL     .-6             ;WAIT FOR PRINTER READY
013202  116777  001726  001646          MOVB    TEMP2,@TPB      ;PRINT CHAR.
013210  012767  000370  000010          MOV     #370,MASK       ;MASK FOR NEXT '5' DIGITS
013216  005367  001710                  DEC     TEMP1
013222  001350                          BNE     SHIFT
013224  000002                          RTI
013226  000376                  MASK:   376

                                        ;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET

013230  105777  001614          TKSFLG: TSTB    @TKS            ;FLAG SET?
013234  100001                          BPL     .+4             ;NO, EXIT
013236  104011                          TTYIN                   ;YES,INQUIRE
013240  000002                          RTI
```

```
                                        ;ENTERED WITH SYSTEM TRAP CALL (ERROR)
                                        ;PRINT OUT THE ERROR PC+2 AND STATUS REGISTER
013242  104016                          LOGERR: TSTTKS                  ;TEST FOR KEYBOARD INTERRUPT
013244  037727  001610  020000                  BIT     @SWR,#20000     ;TEST FOR INHIBIT PRINT OUT
013252  001022                                  BNE     CK              ;INHIBIT CHECK FOR HALT
013254  011667  001646                          MOV     (SP),KSTOR3     ;PC OF FAILING ROUTINE
013260  162767  000002  001640                  SUB     #2,KSTOR3
013266  011167  001636                          MOV     @ADCSR,KSTOR4   ;SAVE A/D STATUS TO BE TYPED
013272  042711  000100                          BIC     #100,@ADCSR     ;CLR INTERRUPT ENABLE
013276  005777  001566                          TST     @ADDBR          ;CLEAR DONE
013302  104000                                  PRINT                   ;OUTPUT CARRIAGE RETURN AND LINE FEED
013304  014045                                  CRLF
013306  104010                                  PRTOCT                  ;PRINT FAILING PC+2
013310  015126                                  KSTOR3
013312  104007                                  SPACE                   ;OUTPUT A SPACE
013314  104010                                  PRTOCT                  ;PRINT PROCESSOR STATUS
013316  015130                                  KSTOR4                  ;CONTENTS OF A/D STATUS REG.
013320  005777  001534                  CK:     TST     @SWR            ;CHECK SR FOR HALT SWITCH
013324  100001                                  BPL     .+4             ;BRANCH IF NOT SET
013326  000000                                  HALT                    ;HALT ON ERROR UP
013330  000002                                  RTI                     ;RETURN TO MAIN LINE

                                        ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 4000 TIMES

013332  104016                          SCOPEC: TSTTKS                  ;TEST FOR KEYBOARD INIT
013334  032777  040000  001516                  BIT     #40000,@SWR     ;TEST SR FOR SCOPE
013342  001012                                  BNE     SCOPEB          ;YES, SCOPE
013344  032777  004000  001506                  BIT     #4000,@SWR      ;NO-TEST FOR ITERATION
013352  001013                                  BNE     SCOPEG          ;INHIBIT ITERATION
013356  026767  000036  000032                  CMP     SCOPEF,ICOUNT   ;COMPARE CURRENT COUNT TO MAX NUMBER
013362  100007                                  BPL     SCOPEG          ;EXIT-DONE
013364  005267  000026                          INC     SCOPEF          ;INCREMENT COUNT
013370  022606                          SCOPEB: CMP     (6)+,SP         ;REPOSITION STACK
013372  012677  001450                          MOV     (6)+,@PSW       ;RESTORE PREVIOUS PROCESSOR STATUS
013376  000177  000016                          JMP     @RETURN         ;REPEAT TEST
013402  005067  000010                  SCOPEG: CLR     SCOPEF          ;CLEAR COUNT
013406  011667  000006                          MOV     @SP,RETURN      ;SAVE SCOPE RETURN POINTER
013412  000002                                  RTI                     ;RETURN INLINE-NEXT TEST
013414  004000                          ICOUNT: 4000                    ;ITERATION COUNT
013416  000000                          SCOPEF: 0                       ;COUNT LOCATION FOR ITERATION LOOP
013420  001636                          RETURN: TEST1+2                 ;ADDRESS OF LAST TEST
```

```
                                                ;MESSAGES
013422      000                                         .BYTE
013423      045  042101  031117  TITLE:  .ASCII  '%AD02/AD11 DIAGNOSTIC TEST, (MAINDEC-11-DZADA-C)@'
013430  040457  030504  020061
013436  044504  043501  047516
013444  052123  041511  052040
013452  051505  026124  024040
013460  040515  047111  042504
013466  026503  030461  042055
013474  040532  040504  041455
013502  040051
013504  040445  042057  046040  MES2:   .ASCII  '%A/D LENGTH? @'
013512  047105  052107  037510
013520  040040
013522  021045     507  047111  MES3:   .ASCII  '%"GAIN ACCURACY TEST". SUPPLY THE FOLLOWING VOLTAGES %'
013530  04044'     1503  051125
013536  0415[     20131  042524
013544  0521[     027042  051440
013552  050125  046120  020131
013560  044124  020105  047506
013566  046114  053517  047111
013574  020107  047526  052114
013602  043501  051505  022440
0'610   052040  020117  042523          .ASCII  " TO SELECTED CH., TYPE 'CR' TO START TEST.@"
013616  042514  052103  042105
013624  041440  027110  020054
013632  054524  042520  023440
013640  051103  020047  047524
013646  051440  040524  052122
013654  052040  051505  027124
013662      100
013663      045  054524  042520  MES4:   .ASCII  "%TYPE LETTER ' ' TO RUN DESIRED TEST:%"
013670  046040  052105  042524
013676  020122  020047  020047
013704  047524  051040  047125
013712  042040  051505  051111
013720  042105  052040  051505
013726  035124      045
013731      047  023514  043517          .ASCII  "'L'OGIC, 'C'ALIBRATION, 'R'EPEATIBILITY, 'G'AIN, "
013736  041511  020054  041447
013744  040447  044514  051102
013752  052101  047511  026116
013760  023440  023522  050105
013766  040505  044524  044502
013774  044514  054524  020054
014002  043447  040447  047111
014010  020054
014012  023522  023505  047503          .ASCII  "R'E'COVERY, 'T'NNN%@"
014020  042526  054522  020054
014026  052047  047047  047116
014034  040045
```

```
014036  041536  040045          CNTRLC: .ASCII  '↑C%@'

014042  040536     100          CNTRLA: .ASCII  '↑A@'

014045     045     100          CRLF:   .ASCII  '%@'

014047     045  040056          DOT:    .ASCII  '%.@'

014052  020077     100          QMARK:  .ASCII  '? @'

014055     045  046042  043517  MES5:   .ASCII  '%"LOGIC TEST"%@'
014062  041511  052040  051505
014070  021124  040045

014074  044502  020124  020062  MES6:   .ASCII  'BIT 2 SET INTR. LEVEL @'
014102  042523  020124  047111
014110  051124  020056  042514
014116  042526  020114     100

014123     045  041442  046101  MES7:   .ASCII  '%"CALIBRATION TEST"%@'
014130  041111  040522  044524
014136  047117  052040  051505
014144  021124  040045
014150  021045  042522  047503  MES8:   .ASCII  '%"RECOVERY TEST"@'
014156  042526  054522  052040
014164  051505  021124     100

014171     045  044103  020056  MES9:   .ASCII  '%CH. @'
014176     100

014177     045  042447  054047  MES10:  .ASCII  "%'E'XT. OR 'I'NT. SYNC? @"
014204  027124  047440  020122
014212  044447  047047  027124
014220  051440  047131  037503
014226  040040

014230  051105  047522  020122  MES12:  .ASCII  'ERROR BIT SET!%@'
014236  044502  020124  042523
014244  020524  040045

014250  021045  042522  042520  MES13:  .ASCII  '%"REPEATIBILITY TEST"@'
014256  052101  041111  046111
014264  052111  020131  042524
014272  052123  040042

014276  022445  044103  024056  MES14:  .ASCII  '%%CH.(S)? @'
014304  024523  020077     100

014311     103  052517  052116  MES16:  .ASCII  'COUNT SPREAD? @'
014316  051440  051120  040505
014324  037504  040040

014330  040507  047111  051450  MES18:  .ASCII  'GAIN(S)? @'
014336  037451  040040
```

```
014342  020045  041440  027110  MES19:  .ASCII  '%  CH.   LO     AV     HI@'
014350  020040  046040  020117
014356  020040  020040  053101
014364  020040  020040  044040
014372  040111

014374  020045  046040  020117  MES20:  .ASCII  '%  LO  -5  -4  -3  -2  -1  AV  +1  +2  +3  +4  +5  HI@'
014402  026440  020065  026440
014410  020064  026440  020063
014416  026440  020062  026440
014424  020061  040440  020126
014432  025440  020061  025440
014440  020062  025440  020063
014446  025440  020064  025440
014454  020065  044040  040111

014462  050045  053517  051105  MES21:  .ASCII  '%POWER FAILURE @'
014470  043040  044501  052514
014476  042522  040040

014502  022445  044103  037456  MES22:  .ASCII  '%%CH.? @'
014510  040040

014512  032453  030056  053060  MES23:  .ASCII  '+5.00V@'
014520     100

014521     123  044527  041524  MES24:  .ASCII  'SWITCH NEG.!@'
014526  020110  042516  027107
014534  040041

014536  031053  032456  053060  MES25:  .ASCII  '+2.50V@'
014544     100

014545     053  027061  032462  MES26:  .ASCII  '+1.25V@'
014552  040126

014554  030053  033056  032462  MES27:  .ASCII  '+0.625V@'
014562  040126

014564  030053  031456  031061  MES28:  .ASCII  '+0.3125V@'
014572  053065     100

014575     053  027060  032461  MES29:  .ASCII  '+0.1563V@'
014602  031466  040126

014606  030053  030056  034067  MES30:  .ASCII  '+0.0781V@'
014614  053061     100

014617     053  027060  031460  MES31:  .ASCII  '+0.0390V@'
014624  030071  040126

014630  043445  044501  020116  MES32:  .ASCII  '%GAIN  5.0000  2.5000  1.2500  0.6250  '
014636  032440  030056  030060
014644  020060  031040  032456
014652  030060  020060  030440
```

```
014660  031056  030065  020060
014666  030040  033056  032462
014674  020060     040
014677     060  031456  031061          .ASCII  '0.3125  0.1563  0.0781  0.390@'
014704  020065  030040  030456
014712  033065  020063  030040
014720  030056  034067  020061
014726  030040  031456  030071
014734     100
014735     045  040507  047111  MES33:  .ASCII  '%GAIN  VOLTAGE AVERAGE%@'
014742  020040  047526  052114
014750  043501  020105  053101
014756  051105  043501  022505
014764     100
014765     045  020061  020040  MES34:  .ASCII  '%1     @'
014772  020040     100
014775     045  020062  020040  MES35:  .ASCII  '%2     @'
015002  020040     100
015005     045  020064  020040  MES36:  .ASCII  '%4     @'
015012  020040     100
015015     045  020070  020040  MES37:  .ASCII  '%8     @'
015022  020040     100
015025     055  026455  026455  MES38:  .ASCII  '------  @'
015032  020055  040040
015036  051445  047131  037503  MES39:  .ASCII  '%SYNC? @'
015044  040040
```

```
                                        .EVEN

                                ;REGISTER ADDRESS

015046  177776                  PSW:    177776                  ;ADDRESS OF PROCESSOR STATUR REG
015050  177560                  TKS:    177560                  ;ADDRESS OF KEYBOARD STATUR REG
015052  177562                  TKB:    177562                  ;ADDRESS OF KEYBOARD BUFFER REG.
015054  177564                  TPS:    177564                  ;ADDRESS OF PRINTER STATUS REG.
015056  177566                  TPB:    177566                  ;ADDRESS OF PRINTER BUFFER REG
015060  177570                  SWR:    177570                  ;ADDRESS OF SWITCH REG.
015062  177571                  SWR0:   177571                  ;ADDRESS OF SWITCH REG. HIGH BYTE.
015064  176770                  ADCS:   176770                  ;A/D STATUS REG ADDRESS
015066  176771                  ADCS0:  176771                  ;HIGH BYTE OF A/D STATUS
015070  176772                  ADDBR:  176772                  ;A/D BUFFER REG.
015072  000130                  ADINT:  130                     ;A/D VECTOR ADDRESS

                                ;ADDRESS AND CONSTANTS TABLE

015074  000000                  ADWRD2: 0                       ;LOW BYTE OF 'ADWORD'
015076  000000                  ADSIGN: 0                       ;BIPOLAR=0, UNIPOLAR=1
015100  000000                  SUBX:   0                       ;LOGIC TEST SUBROUTINE SWITCH
015102  000000                  INTFLG: 0
015104  177634                  DELAY1: -100.                   ;DELAY COUNT FOR LOGIC TEST
015106  001000                  STACK:  1000                    ;INITIAL SP. ADDRESS
015110  001262                  AVECTR: INITA                   ;'↑A' VECTOR ADDRESS
015112  000000                  ADSIZE: 0                       ;OCTAL STORAGE OF A/D LENGTH
015114  000000                  PROC:   0                       ;TEMP STORAGE FOR 'PSW'
015116  000000                  CHRCNT: 0                       ;TEMP STORAGE
015120  000000                  COUNT:  0                       ;TEMP STORAGE
015122  000000                  KSTOR1: 0                       ;PERMANENT STORAGE
015124  000000                  KSTOR2: 0                       ;PERMANENT STORAGE
015126  000000                  KSTOR3: 0                       ;PERMANENT STORAGE
015130  000000                  KSTOR4: 0                       ;PERMANENT STORAGE
015132  000000                  TEMP1:  0                       ;TEMPORARY STORAGE
015134  000000                  TEMP2:  0                       ;TEMPORARY STORAGE
015136  000000                  TEMP3:  0                       ;TEMPORARY STORAGE
015140  000000                  MESPRT: 0
015142  000000                  HIORDV: 0
015144  000000                  HIGH:   0
015146  000000                  LOW:    0
015150  000000                  AVERM5: 0
015152  000000                  AVERM4: 0
015154  000000                  AVERM3: 0
015156  000000                  AVERM2: 0
015160  000000                  AVERM1: 0
015162  000000                  AVRAGE: 0
015164  000000                  AVERP1: 0
015166  000000                  AVERP2: 0
015170  000000                  AVERP3: 0
015172  000000                  AVERP4: 0
015174  000000                  AVERP5: 0
015176  000000                  ORLOW:  0
015200  000000                  MINUS5: 0
015202  000000                  MINUS4: 0
015204  000000                  MINUS3: 0
015206  000000                  MINUS2: 0
```

```
2582  015210  000000                    MINUS1: 0
2583  015212  000000                    AVGCNT: 0
2584  015214  000000                    PLUS1:  0
2585  015216  000000                    PLUS2:  0
2586  015220  000000                    PLUS3:  0
2587  015222  000000                    PLUS4:  0
2588  015224  000000                    PLUS5:  0
2589  015226  000000                    ORHIGH: 0
2590  015230  000000                    XSPRD1: 0
2591  015232  000000                    XSPRD2: 0
2592  015234  000000                    XSPRD3: 0
2593  015236  000000                    XSPRD4: 0
2594  015240  000000                    POS500: 0
2595  015242  000000                    POS250: 0
2596  015244  000000                    POS125: 0
2597  015246  000000                    POS625: 0
2598  015250  000000                    POS312: 0
2599  015252  000000                    NEG500: 0
2600  015254  000000                    NEG250: 0
2601  015256  000000                    NEG125: 0
2602  015260  000000                    NEG625: 0
2603  015262  000000                    NEG312: 0
2604  015264  000000                    GP50X1: 0
2605  015266  000000                    GP25X1: 0
2606  015270  000000                    GP12X1: 0
2607  015272  000000                    GP62X1: 0
2608  015274  000000                    GP31X1: 0
2609  015276  000000                    GP25X2: 0
2610  015300  000000                    GP12X2: 0
2611  015302  000000                    GP62X2: 0
2612  015304  000000                    GP31X2: 0
2613  015306  000000                    GP15X2: 0
2614  015310  000000                    GP12X4: 0
2615  015312  000000                    GP62X4: 0
2616  015314  000000                    GP31X4: 0
2617  015316  000000                    GP15X4: 0
2618  015320  000000                    GP07X4: 0
2619  015322  000000                    GP62X8: 0
2620  015324  000000                    GP31X8: 0
2621  015326  000000                    GP15X8: 0
2622  015330  000000                    GP07X8: 0
2623  015332  000000                    GP03X8: 0
2624  015334  000000                    GM50X1: 0
2625  015336  000000                    GM25X1: 0
2626  015340  000000                    GM12X1: 0
2627  015342  000000                    GM62X1: 0
2628  015344  000000                    GM31X1: 0
2629  015346  000000                    GM25X2: 0
2630  015350  000000                    GM12X2: 0
2631  015352  000000                    GM62X2: 0
2632  015354  000000                    GM31X2: 0
2633  015356  000000                    GM15X2: 0
2634  015360  000000                    GM12X4: 0
2635  015362  000000                    GM62X4: 0
2636  015364  000000                    GM31X4: 0
2637  015366  000000                    GM15X4: 0
```

```
015370  000000                    GM07X4: 0
015372  000000                    GM62X8: 0
015374  000000                    GM31X8: 0
015376  000000                    GM15X8: 0
015400  000000                    GM07X8: 0
015402  000000                    GM03X8: 0

                                  ;HERE STARTS THE '512' WORD A/D DATA BUFFER.
        015400                            .=15400
015400  000000                    ADBUFF: 0

        001000                            .END    INIT
```

```
ADBUFF  015400          1277    1282    1851    1993    2160    2214    2647#
ADCNVT  011446          1274    1331    1766    1840    1843    1992#
ADCS    015064           665    1995*   1997    1999    2537#
ADCSO   015066           768*   1996*   2538#
ADCSR =%000001           518#    665*    681     687*    688     693*    694     705*    707     711*    715     725*    726
                         728*    729     736*    740     744*    748     755*    756*    760     767*    772     779*    781*
                         785     793*    794*    798*    799     806*    807*    811     818*    820*    824     832*    834*
                         835*    836     846     847*    848*    849*    850     853     866*    867     869*    870     878*
                         879     881*    882     885     892*    893     896*    897     900     909*    910*    911     914
                         924*    925*    926*    927     930     938*    939*    940     942*    943     950*    955*    956*
                         957     963*    972*    975*    981*    984     992*    993*    995*    999     1030*   1032*   1034*
                        1036*   1038*   1040*   1049*   1050*   1051    1055*   1060*   1068*   1069*   1070    1075*   1077*
                        1081*   1126*   1128*   1130*   1132*   1134*   1136*   1138*   1142*   1143*   1144    1151*   1159*
                        1164*   1174*   1175    1191*   1192    1195    1202*   1207*   1208*   1209    1222*   2291    2292*
ADDBR   015070           686     691     706     724     735     743     754     766     780     792     805     819     833
                         865     877     891     895     908     923     937     949     974     1048    1067    1126    1128
                        1130    1132    1134    1136    1138    1141    1146    1163    1177    1178    1201    1994    2003
                        2293    2539#
ADINT   015072           666    2540#
ADSIGN  015076           583*   1428    1452    1487    1533    1586    1634    1670    1694    1730    1935*   2545#
ADSIZE  015112           590*    594*    598*    602*    609*    611     616     622*    2164    2172    2183    2189    2193
                        2199    2203    2551#
ADWRD2  015074          1165*   1167*   1168*   1170*   1174    1263*   1265*   1266*   1267*   1273*   1330*   1418*   1758*
                        1786    1791    1796    1832    1835    1838*   1839*   1841*   1842*   1995    1996    2042*   2046*
                        2050*   2054*   2544#
AVECTR  015110           638*   1249*   1306*   1411*   1821*   1921    2550#
AVERM1  015160          1774    2570#
AVERM2  015156          2569#
AVERM3  015154          2568#
AVERM4  015152          2567#
AVERM5  015150          2220    2223    2566#
AVERP1  015164          1772    2190    2572#
AVERP2  015166          2573#
AVERP3  015170          2574#
AVERP4  015172          2575#
AVERP5  015174          2218    2576#
AVGCNT  015212          1383    2240    2583#
AVGDAT  012474          2178#   2181
AVGTAB  010144          1277*   1707*   1743#   1744*   1851*
AVRAGE  015162          1374    1768    1770    1808    2161*   2162    2163    2164*   2173*   2179*   2182*   2183*   2188
                        2198    2200    2571#
BCDBIN  011306          1952#   1984    2078
BCDBN1  011322          1955#   1979
BCDBN2  011326          1957#   1975
BCDEND  011410          1958    1961    1976#
BCDERR  011424          1963    1965    1981#
BCDTAB  011436           588     592     596     600     604     1314    1316    1318    1320    1326    1418    1829    1830
                        1833*   1953    1954*   1985#   2040    2044    2048    2052
BINDEC= 104006           559#   1368    1389    1849
CALBRT  006164           649    1249#
CALBT1  006204          1249    1255#
CALBT2  006240          1263#   1291
CALBT3  006372          1284#   1287
CALBT4  006404          1281    1288#
CALB1A  006210          1254    1257#
CALB2A  006330          1272    1274#   1290
```

```
CALB2B  006360          1276    1282#
CATORI= 104005           558#   1333    1769
CATORZ  012566          2082    2187#
CATR1   012652          2210#   2212
CATR2   012674          2215#   2232    2234    2236
CATR3   012734          2224#   2228
CATR4   012754          2225    2230#
CATR5   013000          2216    2240#
CHKGAN  005654          1166    1169    1171    1173#
CHRCNT  015116          1863*   1893*   1907*   1908    1927    1929*   1957    1959*   1978    2553#
CK      013320          2288    2301#
CMPTE   012334          2081    2158#
CMPUTE= 104004           557#   1332    1767
CNTRLA  014042          1919    2383#
CNTRLC  014036           636    2381#
CNVT1   011472          1996#   2005
CNVT2   011520          2000    2003#
COUNT   015120           712*    713*    737*    738*    745*    746*    757*    758*    769*    770*    782*    783*    795*
                         796*    808*    809*    821*    822*   1173*   1181*   1259*   1329*   1717*   1722*   1727*   1754*
                        1765*   1837*   1992    2554#
CRLF    014045          1126    1128    1130    1132    1134    1136    1138    1261    1280    1359    1366    1386    1737
                        1810    1942    2295    2385#
DASH6   010174          1752#   1755    2090
DECOCT= 104001           554#    587    1313    1325    1417    1828    2039
DECOUT  012234          2129    2135#
DECPNT  012322          2122*   2126    2128    2133*   2150#
DECPRT  012136          2083    2120#
DEC1    012262          2136    2138    2141#
DEC2    012276          2140    2143#
DELAY1  015104           712     737     745     757     769     782     795     808    2548#
DIGCNT  012316          2121*   2130*   2137    2148#
DIGIT   012314          2124*   2125*   2135    2139*   2142*   2145    2147#
DOT     014047           642    2387#
EMTOK   011762          2067    2069#
EMTSRV  011742           542    2063#
EMTTAB  012002          2071    2077#
ERROR = 104400           552#    683     690     698     709     717     731     742     750     762     774     787     801
                         813     826     838     852     872     884     899     913     929     945     959     978     997
                        1030    1032    1034    1036    1038    1040    1058    1079    1126    1128    1130    1132    1134
                        1136    1138    1155    1180    1194    1211    1220
EXTTY   011262          1943#
FILE1   012554          2191#   2195
FILE2   012624          2201#   2205
GAIN    007054           655    1411#
GAINER  010326          1776#
GAININ= 104003           556#   1321    1831
GAINXN= 104017           568#   1834
GANEXT  010336          1771    1773    1775    1778#
GERR1   010340          1777    1779#
GERR2   010370          1782    1786#
GERR3   010404          1787    1791#
GERR4   010422          1792    1796#
GERR5   010440          1797    1801#
GERR6   010444          1790    1795    1800    1803#
GETDAT  012400          2165#   2176
GLOOP   010250          1764#   1780    1811
```

```
GM03X8   015402          1703    2643#
GM07X4   015370          1679    2638#
GM07X8   015400          1685    2642#
GM12X1   015340          1496    2626#
GM12X2   015350          1502    2630#
GM12X4   015360          1507    2634#
GM15X2   015356          1643    2633#
GM15X4   015366          1649    2637#
GM15X8   015376          1655    2641#
GM25X1   015336          1460    2625#
GM25X2   015346          1467    2629#
GM31X1   015344          1595    2628#
GM31X2   015354          1601    2632#
GM31X4   015364          1607    2636#
GM31X8   015374          1613    2640#
GM50X1   015334          1437    2624#
GM62X1   015342          1542    2627#
GM62X2   015352          1547    2631#
GM62X4   015362          1553    2635#
GM62X8   015372          1559    2639#
GP03X8   015332          1693    2623#
GP07X4   015320          1663    2618#
GP07X8   015330          1669    2622#
GP12X1   015270          1475    2606#
GP12X2   015300          1481    2610#
GP12X4   015310          1486    2614#
GP15X2   015306          1621    2613#
GP15X4   015316          1627    2617#
GP15X8   015326          1633    2621#
GP25X1   015266          1445    2605#
GP25X2   015276          1451    2609#
GP31X1   015274          1567    2608#
GP31X2   015304          1573    2612#
GP31X4   015314          1579    2616#
GP31X8   015324          1585    2620#
GP50X1   015264          1427    1707    1743    2604#
GP62X1   015272          1515    2607#
GP62X2   015302          1520    2611#
GP62X4   015312          1526    2615#
GP62X8   015322          1532    2619#
GT0      007066          1411    1414#   1732    1735
GT1      007110          1422#
GT10     010022          1712#   1738
GT11     010114          1731    1733#
GT2      007146          1429    1440#
GT3      007230          1453    1470#
GT4      007326          1488    1510#
GT5      007444          1534    1562#
GT6      007562          1587    1616#
GT7      007660          1635    1658#
GT8      007736          1671    1688#
GT9      007774          1695    1707#
G1     = 000000           512#   1425    1435    1443    1458    1473    1494    1513    1540    1565    1593
G2     = 000010           513#   1449    1465    1479    1500    1518    1545    1571    1599    1619    1641
G4     = 000020           514#   1484    1505    1524    1551    1577    1605    1625    1647    1661    1677
G8     = 000030           515#   1530    1557    1583    1611    1631    1653    1667    1683    1691    1701
```

```
HIGH    015144    1377    2162*   2166    2168*   2564#
HIORDV  015142    2159*   2174*   2178*   2563#
ICOUNT  013414    667*    1187*   1200*   1231*   1262*   1711*   1741    1824*   2313    2322#
INBUF   011272    644     647     650     653     659     1258    1859    1892    1947#   1952
INIT    001000    547     575#    2649
INITA   001262    623#    638     2550
INIT1   001036    583#    608
INIT2   001330    625     638#    664
INIT3   001464    662#    1872
INPUTA  011040    1894#   1895    1913    1928
INPUTB  011110    1903    1906#   1938
INTFLG  015102    670*    1126*   1128*   1130*   1132*   1134*   1136*   1138*   2547#
INTVC =%000004    519#    666*    952*    953*    960     964*    969*    971*    982*    990*    991*    1001*   1030*
                  1032*   1034*   1036*   1038*   1040*   1053*   1054*   1061*   1072*   1073*   1083*   1126*   1128*
                  1130*   1132*   1134*   1136*   1138*   1147*   1149*   1157*   1204*   1205*   1212    1215*   1218
                  1223*
KSTOR1  015122    1268*   1269*   1283    1314*   1315    1318    1327    1463*   1708*   1733*   1829*   1838    1860*
                  1862*   1865*   1866*   1867*   1870    1873*   1876    2555#
KSTOR2  015124    1270*   1289    1315*   1320*   1393    1762*   1768*   1830*   1841    1848    2556#
KSTOR3  015126    1326*   1338    1343    1349    1354    1832*   1839    2289*   2290*   2297    2557#
KSTOR4  015130    1327*   1330    1367    1392*   1393    1835*   1842    2291*   2300    2558#
LDNEG   001240    618#    621
LDPOS   001210    612#    615
LDSIZE  001176    591     595     599     603     610#
LOGERR  013242    544     2286#
LOGIC   001532    646     675#
LOW     015146    1371    2163*   2169    2171*   2565#
LPTEST  006160    1233    1239#
MASK    013226    2261*   2267    2272*   2276#
MESPRT  015140    1309*   1360    1378    1382*   1419*   1781    1783*   2562#
MES10   014177    1253    2411#
MES12   014230    2002    2418#
MES13   014250    1308    2422#
MES14   014276    1311    1826    2427#
MES16   014311    1323    2431#
MES18   014330    2037    2435#
MES19   014342    1363    2438#
MES2    013504    585     2338#
MES20   014374    1381    2444#
MES21   014462    2031    2454#
MES22   014502    1415    2458#
MES23   014512    1423    2461#
MES24   014521    1433    1456    1492    1538    1591    1639    1675    1699    2464#
MES25   014536    1441    2468#
MES26   014545    1471    2471#
MES27   014554    1511    2474#
MES28   014564    1563    2477#
MES29   014575    1617    2480#
MES3    013522    1413    2342#
MES30   014606    1659    2483#
MES31   014617    1689    2486#
MES32   014630    1710    2489#
MES33   014735    1785    2503#
MES34   014765    1713    1789    2509#
MES35   014775    1716    1794    2512#
MES36   015005    1721    1799    2515#
```

```
MES37   015015          1726    1802    2518#
MES38   015025          1753    2521#
MES39   015036          1256    2523#
MES4    013663           624    2360#
MES5    014055           676    2391#
MES6    014074          1126    1128    1130    1132    1134    1136    1138    2395#
MES7    014123          1251    2400#
MES8    014150          1823    2404#
MES9    014171          1847    2408#
MINUS1  015210          2242    2582#
MINUS2  015206          2245    2581#
MINUS3  015204          2248    2580#
MINUS4  015202          2251    2579#
MINUS5  015200          2231*   2578#
MONITR  001270           548     628#   1924    2032
NEG125  015256          1495    1546    1606    1654    2601#
NEG250  015254          1459    1501    1552    1612    2600#
NEG312  015262           620    1594    1642    1678    1702    2603#
NEG500  015252           614    1436    1466    1506    1558    2599#
NEG625  015260          1541    1600    1648    1684    2602#
NEWIN   011034          1893#   1946
NINT    005562          1147    1155#
NOP   = 000240           526#
OCTPRT  013112          2085    2257#
ORHIGH  015226          2211    2233*   2589#
ORLOW   015176          1387    2209    2235*   2577#
OUTPTA  011126          1905    1910#   1911    1932    1936
OVRHI   012764          2219    2233#
OVRLO   012772          2221    2235#
PC    =%000007           508#   1235*   1274*   1331*   1766*   1840*   1843*   1858*   2006*   2129*
PLUS1   015214          2241    2584#
PLUS2   015216          2244    2585#
PLUS3   015220          2247    2586#
PLUS4   015222          2250    2587#
PLUS5   015224          2588#
POPRO = 012600           523#
POP1SP= 005726           521#
POP2SP= 022626           525#   1030    1032    1034    1036    1038    1040    1920
POS125  015244          1474    1519    1578    1632    2596#
POS250  015242          1444    1480    1525    1584    2595#
POS312  015250          1566    1620    1662    1692    2598#
POS500  015240           610    1426    1450    1485    1531    2594#
POS625  015246          1514    1572    1626    1668    2597#
PRINT = 104000           553#    581     584     606     623     635     641     662     675    1126    1128    1130    1132
                        1134    1136    1138    1250    1252    1255    1260    1279    1307    1310    1322    1358    1362
                        1365    1380    1385    1412    1414    1422    1432    1440    1455    1470    1491    1510    1537
                        1562    1590    1616    1638    1658    1674    1688    1698    1709    1712    1715    1720    1725
                        1736    1752    1784    1788    1793    1798    1801    1809    1822    1825    1846    1918    1941
                        1944    1981    2001    2030    2036    2294
PROC    015114           710*    720     830*    839     844*    855     863*    873     906*    916     921*    932    1258*
                        1271    2016*   2021    2552#
PRTADR  010446          1760*   1770    1772    1774    1804#
PRTAVG= 104014           565#   1278    1714    1719    1724    1729    1852
PRTOCT= 104010           561#   1126    1128    1130    1132    1134    1136    1138    1370    1373    1376    1742    1803
                        1807    2296    2299
PSW     015046           576*    630*    669*    692*    710     720*    830     831*    839*    844     845*    855*    863
```

```
                          864*    873*    906     907*    916*    921     922*    932*    951*    953     954*    960*    970*
                          971     973*    977*    989*    991     1030*   1032*   1034*   1036*   1038*   1040*   1047*   1054
                         1066*   1073    1074*   1078*   1126*   1128*   1130*   1132*   1134*   1136*   1138*   1148*   1149
                         1150*   1153*   1203*   1205    1206*   1212*   1216*   1218*   2020*   2099*   2120*   2257*   2317*
                         2530*
PUSHRO= 010046            522*
PUSH1S= 005746            520*
PUSH2S= 024646            524*
PWRFAL  011534            536     2009*
PWRUP   011570           2017     2020*
QMARK   014052            607      663    1945    1982    2389*
RECVRY  010466            661     1821*
RECVY1  010506           1821     1825*
RECVY2  010560           1836*    1845    1853
RELOOP  005676           1178*    1182
REPTST  006420            652     1306*
REPT1   006432           1306     1309*   1319
REPT2   006506           1317     1321*
REPT2A  006526           1327*    1395
REPT3   006534           1328*    1394
REPT4   006710           1335     1341    1346    1352    1355    1358*
REPT5   006726           1361     1364*
REPT6   006776           1379     1382*
REPT6A  007024           1388*    1391
REPT7   007036           1337     1342    1347    1353    1357    1384    1392*
RESTRT  001536            579      633     677*   1239
RETURN  013420            668*    1230*   1875*   2318    2320*   2324*
R0     =%000000           501*     610*    612*    614     618*    620     1177*   1178    1282*   1967*   1974    2009    2028*
                         2214*    2217
R1     =%000001           502*    1232*   1235    1859*   1861*   1862    1896*   1897*   1898    1900    1902    1906    1912
                         1916     1922    1925    1930*   1933    1937    1939    1955*   1968    1969*   1970*   1971*   1972*
                         1973*    1974*   1977    2010    2027*   2222*   2226*   2227    2230*   2231*
R2     =%000002           503*     694*    695*    696     1367*   1388*   1848*   1956*   1968*   1972    1973    2011    2026*
                         2223*    2224
R3     =%000003           504*    1953*   1977*   2012    2025*   2190*   2192*   2193*   2200*   2202*   2203*   2209*   2210*
                         2211
R4     =%000004           505*    1387*   1388    1390    1892*   1906*   1931    1952*   1960    1962    1964    1966*   1967
                         1976     1993*   2003*   2013    2024*   2160*   2161    2165
R5     =%000005           506*    2014    2023*   2100*   2104    2107    2109    2115    2258*   2263*   2264*   2265*   2266
SCOPE = 104002            555*     684     701     722     733     752     764     776     789     803     815     828     841
                           857     875     888     903     918     934     947     967     987    1030    1032    1034    1036
                         1038     1040    1046    1065    1126    1128    1130    1132    1134    1136    1138    1140    1162
                         1184     1198    1225
SCOPEB  013370           2310     2316*
SCOPEC  013332           2079     2308*
SCOPEF  013416           1874*    2313    2315*   2319*   2323*
SCOPEG  013402           2312     2314    2319*
SETUP   001472            665*     677    1858
SET1    005462           1138*
SET2    005310           1136*
SET3    005136           1134*
SET4    004764           1132*
SET5    004612           1130*
SET6    004440           1128*
SET7    004266           1126*
SHIFT   013144           2263*    2274
```

```
SIXDSH= 104015          566#    1718    1723    1728
SIZE11  001072          589     592#
SIZE12  001112          593     596#
SIZE13  001132          597     600#
SIZE14  001152          601     604#
SIZ14A  001170          605     609#
SP    =%000006          507#    575*    629*    1059    1082    1126    1128    1130    1132    1134    1136    1138    1156
                       1758    1759*   1760    1761*   1762    1763*   2009*   2010*   2011*   2012*   2013*   2014*   2015*
                       2016    2021*   2022    2023    2024    2025    2026    2027    2028    2063*   2064*   2065*   2066
                       2069*   2070*   2071*   2072*   2073    2100    2101*   2258    2259*   2289    2316    2320
SPACE = 104007          560#    1369    1372    1375    1746    1805    1806    1850    2298
SPACEX  011026         1745*   1885*   1887*   1889#
SPCHR   011142         1899    1901    1916#
SPCHR1  011162         1917    1922#
SPCHR2  011224         1926    1933#
SPCHR3  011242         1934    1937#
SPCHR4  011250         1933#
SPCHR5  011264         1909    1940    1944#
STACK   015106          575     629    2549#
SUBX    015100          640*    656    1904*   2546#
SWR     015060         1266    1268    1270    1275    1289    1334    1336    1776    1779    1844    2287    2301    2309
                       2311    2535#
SWR0    015062         1263    2536#
SW00  = 000001          497#
SW01  = 000002          496#
SW02  = 000004          495#
SW03  = 000010          494#
SW04  = 000020          493#
SW05  = 000040          492#
SW06  = 000100          491#
SW07  = 000200          490#
SW08  = 000400          489#
SW09  = 001000          488#
SW10  = 002000          487#    1334
SW11  = 004000          486#
SW12  = 010000          485#
SW13  = 020000          484#    1336    1779    1844
SW14  = 040000          483#    1776
SW15  = 100000          482#
TAGA    012440         2170    2172#
TAKEGN= 104013          564#    1448    1464    1478    1483    1499    1504    1517    1523    1529    1544    1550    1556
                       1570    1576    1582    1598    1604    1610    1624    1630    1646    1652    1666    1682
TEMP1   015132          611*    612     613*    616*    617*    618     619*    1992*   2004*   2158*   2175*   2177*   2180*
                       2187*   2194*   2197*   2204*   2213*   2215*   2260*   2273*   2559#
TEMP2   015134         1283*   1286*   2165*   2166    2168    2169    2171    2172*   2173    2188*   2189*   2191*   2192
                       2198*   2199*   2201*   2202    2217*   2218    2220    2224    2266*   2267*   2268*   2271    2560#
TEMP3   015136         1741*   1747*   2561#
TESTX   010666          658    1858#
TEST0   001560          668     684#   1230
TEST1   001634          701#   2324
TEST10  002306          825     828#
TEST11  002352          841#
TEST12  002420          857#
TEST13  002470          875#
TEST14  002522          888#
TEST15  002560          903#
```

```
TEST16  002630          918#
TEST17  002700          934#
TEST18  002730          947#
TEST19  003030          967#
TEST2   001712          722#
TEST20  003126          987#
TEST21  003204         1030#
TEST22  003276         1032#
TEST23  003370         1034#
TEST24  003462         1036#
TEST25  003554         1038#
TEST26  003646         1040#
TEST27  003740         1046#
TEST28  004024         1065#
TEST29  004126         1126#
TEST3   001740          733#
TEST30  004300         1126    1128#
TEST31  004452         1128    1130#
TEST32  004624         1130    1132#
TEST33  004776         1132    1134#
TEST34  005150         1134    1136#
TEST35  005322         1136    1138#
TEST36  005474         1138    1140#
TEST37  005600         1162#
TEST38  005716         1172    1184#
TEST39  005752         1198#
TEST4   002022          752#
TEST5   002056          764#
TEST6   002114          776#
TEST7   002152          789#
TEST8   002214          803#
TEST9   002252          815#
TINTD   003014          952     962#
TINTE   003106          969     980#
TINTX   004112         1072    1081#
TINT0B  005416         1138#
TINT1A  003724         1040#
TINT1B  005244         1136#
TINT2   006044         1204    1214#
TINT2A  003632         1038#
TINT2B  005072         1134#
TINT3   006070         1213    1215    1219    1220#
TINT3A  003540         1036#
TINT3B  004720         1132#
TINT4A  003446         1034#
TINT4B  004546         1130#
TINT5A  003354         1032#
TINT5B  004374         1128#
TINT6A  003262         1030#
TINT6B  004222         1126#
TINT7A  003164          990     997#
TINT7B  004006         1053    1058#
TITLE   013423          582    2328#
TKB     015052         1896    2532#
TKS     015050          580*    634*    639*    1894    2280    2531#
TKSFLG  013230         2091    2280#
```

```
TPB      015056          1229*   1884*   1912*   2109*   2111*   2114*   2145*   2271*   2534*
TPS      015054           678    1188    1227    1882    1910    2102    2112    2143    2269    2533*
TSLO     012422          2167    2169*
TSTA     010702          1861*   1868
TSTCT1   006670          1350    1354*
TSTCT2   006646          1344    1349*
TSTCT3   006624          1339    1343*
TSTCT4   006572          1336*
TSTDN    005754          1200*   1870
TSTTKS=  104016           567*   1284    1288    1328    1364    1764    1836    2286    2308
TSTXB    010736          1864    1870*
TTYIN =  104011           562*    586     643    1257    1312    1324    1416    1757    1827    1983    2038    2282
TYPECL   012110          2108    2111*
TYPERA   012056          2102*   2103    2110    2116
TYPER1   012074          2105    2107*
TYPER2   012102          2109*
TYPMES   012042          2077    2099*
TYPT1    012164          2124*   2134
TYPT2    012172          2125*   2127
TYPT3    012224          2131    2133*
WAITGN=  104012           563*   1424    1434    1442    1457    1472    1493    1512    1539    1564    1592    1618    1640
                         1660    1676    1690    1700
XGAINA   011634          2036*   2053    2080
XGAINB   011644          2040*   2092
XGAIN2   011662          2041    2044*
XGAIN4   011702          2045    2048*
XGAIN8   011722          2049    2052*
XLOOP    010760          1874*   1875
XPRTAV   010134          1741*   2089
XPTA1    010142          1742*   1748
XSPACE   010776          1882*   1886    2084
XSPRD1   015230          1356    1390    2240*   2241*   2242*   2243    2590*
XSPRD2   015232          1351    2243*   2244*   2245*   2246    2591*
XSPRD3   015234          1345    2246*   2247*   2248*   2249    2592*
XSPRD4   015236          1340    2249*   2250*   2251*   2593*
XTTYIN   011030           539    1892*   2086
XWATGN   010210          1757*   2087    2088
ZERO     012320          2123*   2139    2141*   2149*
.      = 015402           529*    534     535*    538*    541*    546*    574*    578     632     645     648     651     654
                          657     660     679     682     689     697     708     714     716     727     730     739     741
                          747     749     759     761     771     773     784     786     797     800     810     812     823
                          837     851     854     868     871     880     883     886     894     898     901     912     915
                          928     931     941     944     958     961     979     985     996    1000    1030    1032    1034
                         1036    1038    1040    1052    1057    1071    1080    1126    1128    1130    1132    1134    1136
                         1138    1145    1154    1176    1179    1189    1193    1196    1210    1228    1734    1871    1883
                         1923    1948*   1998    2113    2144    2150    2262    2270    2281    2302    2646*
```

INT	1007#	1029	1031	1033	1035	1037	1039	
INT2	1086#	1125	1127	1129	1131	1133	1135	1137

ADC	2174	2182													
ADD	1744	1759	1761	1763	1862	1873	1972	1973	1974	2071	2101	2128	2133	2164	2172
	2173	2189	2199	2241	2242	2244	2245	2247	2248	2250	2251	2259			
ASL	622	1865	1866	1867	1969	1970	1971	2069	2230						
ASR	613	619	1463	2178											
BEQ	578	605	632	657	682	689	697	1179	1233	1290	1317	1357	1384	1429	1453
	1488	1534	1587	1635	1671	1695	1771	1773	1775	1777	1864	1928	1938	1961	2108
	2138	2216	2225												
BGE	1394														
BGT	1886	1965	2005	2170											
BIC	695	831	845	864	907	922	954	973	1074	1150	1170	1206	1269	1861	1897
	1966	2070	2292												
BICB	798	1267	2267												
BIS	692	951	970	977	989	1030	1032	1034	1036	1038	1040	1047	1066	1078	1126
	1128	1130	1132	1134	1136	1138	1148	1153	1167	1203	1273	1382	1783	2142	2268
BISB	779	820													
BIT	688	1334	1336	1776	1779	1844	2287	2309	2311						
BITB	2269														
BLE	1958	2167													
BLT	1963														
BMI	716	730	761	773	800	812	837	851	871	883	912	928	1319	1899	1901
	1909	2219	2221												
BNE	589	593	597	601	615	621	645	648	651	654	660	714	739	747	759
	771	784	797	810	823	1126	1128	1130	1132	1134	1136	1138	1182	1272	1335
	1337	1339	1341	1344	1346	1350	1352	1355	1361	1379	1391	1731	1734	1748	1755
	1780	1782	1787	1792	1797	1845	1903	1917	1923	1926	1934	1940	1979	2041	2045
	2049	2053	2067	2105	2131	2136	2176	2181	2195	2205	2212	2228	2274	2288	2310
	2312														
BPL	679	708	727	741	749	786	825	854	868	880	886	894	898	901	915
	931	941	944	958	985	1000	1030	1032	1034	1036	1038	1040	1052	1071	1126
	1128	1130	1132	1134	1136	1138	1145	1176	1189	1193	1196	1210	1228	1276	1287
	1871	1883	1895	1911	1998	2000	2103	2113	2127	2144	2270	2281	2302	2314	
BR	591	595	599	603	608	625	664	961	979	996	1030	1032	1034	1036	1038
	1040	1057	1080	1126	1128	1130	1132	1134	1136	1138	1154	1172	1213	1219	1254
	1281	1291	1342	1347	1353	1395	1738	1790	1795	1800	1811	1853	1868	1905	1913
	1932	1936	1946	1975	2110	2116	2134	2140	2232	2234	2236	2262			
CLC	1264														
CLR	583	640	670	687	756	781	821	834	835	848	881	926	942	965	983
	1002	1030	1032	1034	1036	1038	1040	1050	1062	1069	1084	1126	1128	1130	1132
	1134	1136	1138	1143	1158	1165	1216	1224	1309	1419	1860	1874	1887	1893	1954
	1955	1956	2099	2120	2159	2210	2222	2257	2319						
CLRB	705	728	736	744	755	767	768	793	818	832	847	924	938	950	963
	972	981	992	995	1030	1032	1034	1036	1038	1040	1049	1060	1068	1077	1081
	1126	1128	1130	1132	1134	1136	1138	1142	1159	1164	1202	1222	2042		
CMP	588	592	596	600	604	614	620	659	696	962	980	998	1059	1082	1126
	1128	1130	1132	1134	1136	1138	1156	1178	1214	1221	1271	1289	1318	1338	1340
	1343	1345	1349	1351	1354	1356	1383	1390	1393	1770	1772	1774	1870	1908	1964
	2040	2044	2048	2052	2137	2166	2169	2211	2218	2220	2224	2227	2313	2316	
CMPB	644	647	650	653	1791	1796	1898	1900	1902	1916	1922	1925	1933	1937	1939
	1960	1962	2104	2107											
DEC	1181	1286	1733	1747	1754	1863	1885	1929	1959	2004	2175	2180	2194	2201	2204
	2215	2273													
EMT	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567
	568														
HALT	534	2018	2068	2229	2303										
INC	713	738	746	758	770	783	796	807	809	849	910	1191	1392	1907	2125

	2130	2191	2226	2231	2233	2235	2315								
INCB	711	822	869	896	925										
JMP	547	548	579	633	646	649	652	655	658	661	1239	1732	1735	1872	1876
	1921	1924	1984	2032	2073	2318									
JSR	677	1166	1169	1171	1235	1274	1331	1766	1840	1843	1858	2129			
MOV	575	576	580	590	594	598	602	609	610	611	612	616	618	629	630
	634	638	639	665	666	667	668	669	693	694	710	712	720	725	737
	745	757	769	782	794	795	808	830	839	844	855	863	866	873	878
	892	906	909	916	921	932	939	952	953	955	960	964	969	971	975
	982	990	991	993	1001	1030	1032	1034	1036	1038	1040	1053	1054	1061	1072
	1073	1083	1126	1128	1130	1132	1134	1136	1138	1147	1149	1157	1173	1174	1177
	1187	1200	1204	1205	1207	1212	1215	1218	1223	1229	1230	1231	1232	1249	1258
	1259	1262	1268	1270	1277	1282	1283	1306	1314	1315	1320	1326	1327	1329	1367
	1387	1388	1411	1707	1708	1711	1717	1722	1727	1741	1745	1760	1762	1765	1768
	1821	1824	1829	1830	1833	1837	1848	1851	1859	1875	1884	1892	1896	1904	1906
	1930	1935	1952	1953	1967	1968	1977	1992	1993	2003	2009	2010	2011	2012	2013
	2014	2015	2016	2017	2020	2021	2022	2023	2024	2025	2026	2027	2028	2063	2065
	2072	2100	2111	2114	2121	2122	2123	2124	2139	2141	2145	2158	2160	2161	2162
	2163	2165	2168	2171	2177	2187	2188	2190	2192	2197	2198	2200	2202	2209	2213
	2214	2217	2223	2240	2243	2246	2249	2258	2260	2261	2272	2289	2291	2317	2320
MOVB	806	956	1055	1075	1126	1128	1130	1132	1134	1136	1138	1151	1168	1208	1263
	1266	1330	1418	1758	1832	1835	1838	1839	1841	1842	1912	1995	1996	2046	2050
	2054	2109	2266	2271											
NEG	617														
NOP	976	994	1056	1076	1126	1128	1130	1132	1134	1136	1138	1152	1217	1236	1237
	1238														
RESET	628	680	1190	1234	1285	2029									
ROL	2263	2264	2265												
ROR	2179														
RORB	1265														
RTI	1749	1756	1778	1888	1943	1980	2043	2047	2051	2055	2106	2132	2184	2252	2275
	2283	2304	2321												
RTS	671	1183	2006	2146											
SUB	2064	2126	2183	2193	2203	2290									
TRAP	552														
TST	577	631	681	686	691	706	724	735	743	754	766	780	792	805	819
	833	836	850	865	870	877	882	891	895	897	908	911	923	927	937
	943	949	974	1048	1067	1126	1128	1130	1132	1134	1136	1138	1141	1146	1163
	1192	1316	1360	1378	1428	1452	1487	1533	1586	1634	1670	1694	1730	1781	1927
	1931	1957	1976	1978	1994	1999	2066	2135	2293	2301					
TSTB	656	678	707	715	726	729	740	748	760	772	785	799	811	824	846
	853	867	879	885	893	900	914	930	940	957	984	999	1030	1032	1034
	1036	1038	1040	1051	1070	1126	1128	1130	1132	1134	1136	1138	1144	1175	1188
	1195	1201	1209	1227	1275	1786	1882	1894	1910	1997	2102	2112	2115	2143	2280
.ABS	474														
.ASCII	2328	2338	2342	2351	2360	2367	2376	2381	2383	2385	2387	2389	2391	2395	2400
	2404	2408	2411	2418	2422	2427	2431	2435	2438	2444	2454	2458	2461	2464	2468
	2471	2474	2477	2480	2483	2486	2489	2496	2503	2509	2512	2515	2518	2521	2523
.BYTE	2327														
.END	2649														
.EVEN	2526														
.MACR	1007	1086													
.REM	1														
.REPT	530														
.TITLE	473														

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZADAC/SOL/CRF/PAGNUM=DZADAC
RUN-TIME: 9 17 3 SECONDS
RUN-TIME RATIO: 85/30=2.7
CORE USED:  10K  (19 PAGES)
```

N06

Spooler runtime 9 Seconds, 42 KCS, 259 disk reads, 4 disk writes, 77 pages

[redacted]

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011111111111111111111111111111110
0000000011111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000001111111112222222222333123
[redacted]56789012345678901 2 *
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011111111111111111111111111111110
0000000011111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000001111111112222222222333123
```

```
  10                    ...B1    2103  012062  100375   ...B5
  43                    ...C1    2156                   ...C5
  97                    ...D1    2194  012570  005367   ...D5
                        ...E1    2246  013044  016767   ...E5
 155                    ...F1    2293  013276  005777   ...F5
                        ...G1    2334  013466  026503   ...G5
 213                    ...H1    2390  014052  020077   ...H5
 267                    ...I1    2446  014410  020064   ...I5
 318                    ...J1    2502  014734     100   ...J5
 372                    ...K1    2535  015060  177570   ...K5
                        ...L1    2591  015232  000000   ...L5
 431                    ...M1    2647  015400  000000   ...M5
 487            002000  ...N1                           ...N5

                        ...B2  CHRCNT  015116           ...B6
 544  000034  013242    ...C2  GM25X1  015336           ...C6
 578  001016  001402    ...D2  INPUTA  011040           ...D6
                        ...E2  MINUS1  015210           ...E6
 635  001324  104000    ...F2  PWRUP   011570           ...F6
 674                    ...G2                           ...G6
 708  001646  100001    ...H2  TEST24  003462           ...H6
 761  002052  100401    ...I2  TSTTKS= 104016           ...I6
 812  002246  100401    ...J2                           ...J6
 866  002442  012711    ...K2  BGT       1886    1965   ...K6
 912  002612  100401    ...L2             982     990   ...L6
 956  002772  112711    ...M2                           ...M6
 996  003162  000402    ...N2   **END** USER DAVIES,TOM ...N6

                        ...B3
                        ...C3
                        ...D3
1050  003756  005011    ...E3
1094                    ...F3
1133  004776            ...G3
1189  005732  100375    ...H3
1207  006016  012711    ...I3
1249  006164  012767    ...J3
1301                    ...K3
1352  006664  001011    ...L3
1405                    ...M3
1456  007202  014521    ...N3

1507  007324  015360    ...B4
1558  007440  015252    ...C4
1612  007556  015254    ...D4
1666  007674  104013    ...E4
1713  010024  014765    ...F4
1766  010260  004767    ...G4
1822  010474  104000    ...H4
1863  010712  005367    ...I4
1899  011062  100427    ...J4
1953  011312  012703    ...K4
1998  011504  100375    ...L4
2042  011654  105067    ...M4
2066  011754  005716    ...N4
```