ADF11

ANALOG TESTS

MD-11-DZADH-A

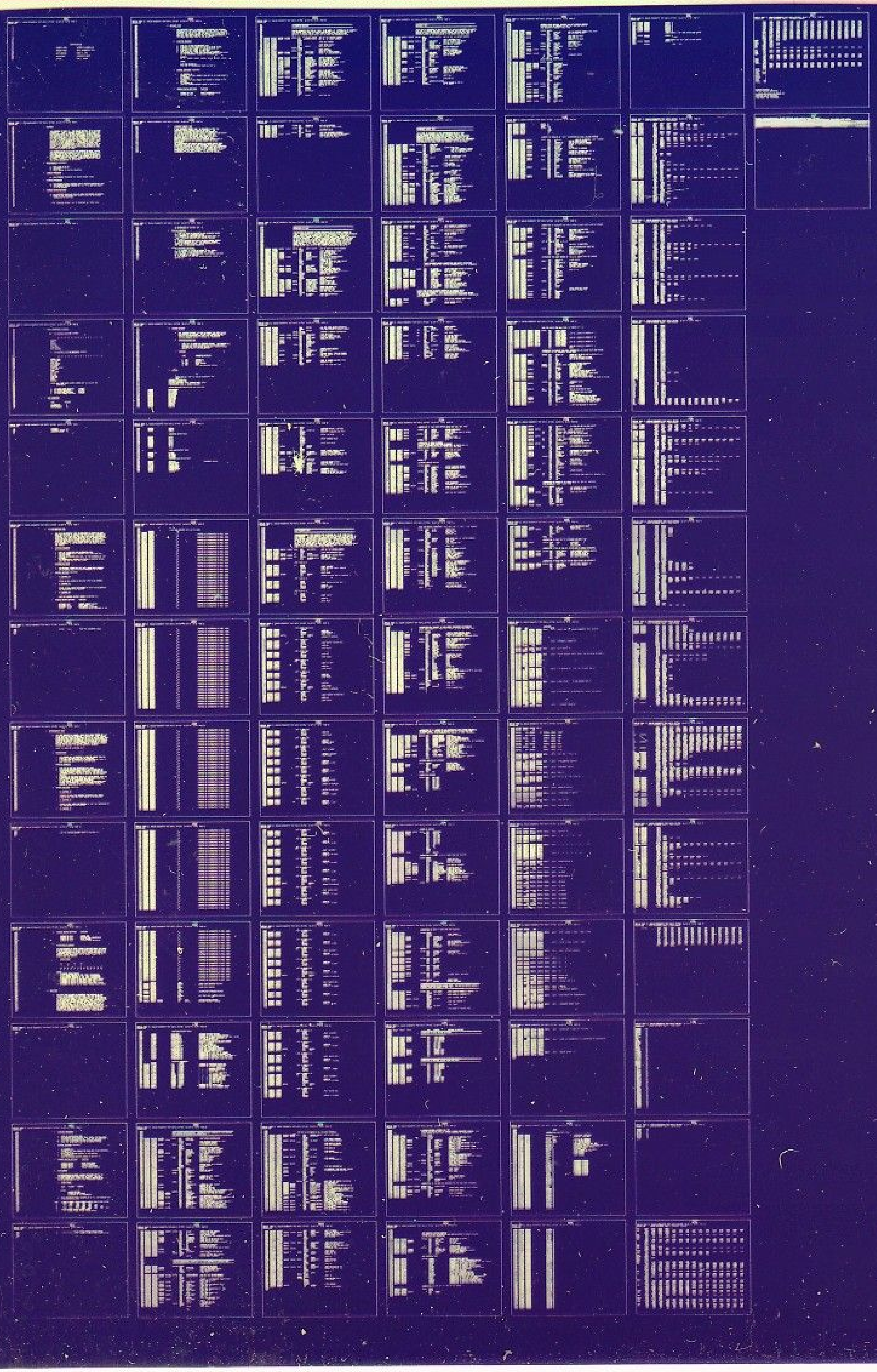
EP-DZADH-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A



```
                        .REM    %












                                                          IDENTIFICATION

                                          PRODUCT CODE:            MAINDEC-11-DZADH-A-D

                                          PRODUCT NAME:            ADF11 DIAGNOSTIC TEST

                                          DATE CREATED:            MARCH 4, 1974

                                          MAINTAINER:              DIAGNOSTIC GROUP

                                          AUTHOR:                  EARL L. BOUSE
```

1. ABSTRACT

THIS PROGRAM IS PART II, THE ANALOG SECTION, OF A TWO PART DIAGNOSTIC. PART I (MAINDEC 11-DZADB-A) IS THE LOGIC TEST SECTION AND SHOULD BE RUN AND PROVED FULLY OPERATIONAL BEFORE RUNNING THIS TEST SECTION.THE PROGRAM IS SELF STARTING AND WHEN LOADED WILL TYPE OUT THE PROGRAM TITLE AND REQUEST FOR THE A/D LENGTH TO BE TYPED.THE PROGRAM WILL ACCEPT A 10 TO 15 BIT UNIPOLAR OR BIPOLAR INPUT. EXAMPLE: 10(CR)* WOULD INDICATE A 10 BIT UNIPOLAR A/D; WHERE TYPING 10+(CR) WOULD INDICTE A 10 BIT BIPOLAR A/D. A SENTANCE IS THEN TYPED GIVING THE LETTER DESIGNATORS TO BE TYPED TO RUN ANY ONE OF THE FOUR (4) SEPERATE TEST OF WHICH THIS PROGRAM IS COMPRISED. THE PROGRAM THEN WAITS IN A KEYBOARD MONITOR MODE FOR A TEST LETTER TO BE TYPED.

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 SIMULANTEOUSLY WHILE RUNNING ANY TEST WILL ENABLE THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR AND AWAITS 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. ADF11 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.

```
    81                                                    -1-
```

```
                        6.  ADF11 ADDRESSES & FORMATS
                            -------------------------

                                A.  A/D CONTROL REGISTER (164006)
                                    -------------------------

                                        15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0

                                SEQ/RAN
                                GAIN
                                DMA/PC
                                EXT/INT
                                INC. MEM
                                FINAL/INIT.
                                MUX. ADDRESS
                                B.  A/D CONTROL & STATUS REGISTER (164010)
                                    ---------------------------------

                                        15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0

                                ERROR & NXM
                                CNV IN PROG.
                                INIT A/D
                                DATA OVRFLO
                                CLR FLAGS
                                A/D WAIT
                                FINAL CH. FLAG
                                W.C. FLAG
                                READY
                                INTR. ENA.
                                EA <17>
                                EA <16>
                                XFER CHK
                                XFER ERR
                                BURST
                                GO/STOP

                                *NOTE: ALL 'CSR' BITS EXCEPT; ERROR,INIT & A/D WAIT ARE
                                       READ/WRITE.


                                C.  A/D WORD COUNT REGISTER      (164004)
                                D.  A/D STATUS REGISTER          (164000)
                                E.  A/D DATA WORD REGISTER       (164012)
                                F.  A/D WORD ADDRESS REGISTER    (164002)

                        7.  TEST DIRECTORY
                            --------------

                                TEST            SECTION
                                ----            -------
                                CALIBRATION       8.
                                REPEATIBILITY     9.
                                GAIN ACCURACY     10.
```

```
                                                RECOVERY         11.
                                                INCREMENT MEMORY 12.
                                                COMMAND DECODER  13.
                                                                       -2-
```

```
8.  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 CONVERTED
        VALUE MAY BE PLACED IN R0 AND DISPLAYED IN THE DATA LIGHTS
        WITH SWITCHES '9-11' WHICH ISSUES RESETS OR PRINTED ON
        THE TELEPRINTER. IF THE SWITCHES '9-11' ARE DOWN, NO RESETS
        ARE ISSUED.

    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.

        3.  ↑P (CONTROL P)
            --------------

            LOAD THE COMMAND DECODER (REFER TO SECTION 13.)

    E.  CONSOLE SWITCH SETTINGS      FUNCTION
        -----------------------      --------

            SWITCHES '0-8'           CHANNEL SELECT (0-777)
            SWITCHES '9-11'          RESET COUNT (0-7)
            SWITCHES '13-14'         GAIN SELECT (1,2,4,8)
            SWITCH   '15=0'          CONVERSION VALUE DISPLAYED IN 'R0'
```

H01

```
   198                                          SWITCH   '15=1'          PRINT THE CONVERTED VALUE
   199
   200
   201                                                     -3-
```

9. REPEATIBILITY TEST

A. THE TEST REQUESTS A CHANNEL, GAIN AND A COUNT SPREAD OF '1-4' TO BE TYPED IN BY THE OPERATOR. A SERIES OF '1024' 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. 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. THE PROGRAM AUTOMATICALLY DOES A CORE SIZING AND IF THERE IS SUFFICIENT MEMORY AVAILABLE TO SUPPORT THE 'INCREMENT MEMORY' FEATURE OF THE A/D, THE USER IS GIVEN THE OPTION OF USING THIS AS A MEANS OF TESTING REPEATIBILITY.
3. A REQUEST IS THEN MADE FOR CH.(S) TO BE TESTED, GAIN AND COUNT SPREAD (RANGE IN WHICH ALL 1024 COUNT MUST FALL FOR THE CH. TO BE CONSIDERED ACCEPTABLE).
4. 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 C)

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

3. ↑P (CONTROL P)

```
                                        LOAD THE COMMAND DECODER (REFER TO SECTION 13.)
                                                        -4-
```

```
        E.  CONSOLE SWITCH SETTINGS     FUNCTION
            -----------------------     --------

                CONSOLE SW 12=0         NORMAL RUN
                CONSOLE SW 12=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. IT SHOULD BE NOTED THAT THIS MAY NOT BE
            A TRUE REPRESENTATION OF ALL '1024' COUNTS WHEN USING THE
            'INCREMENT MEMORY' FEATURE SINCE NO ATTEMPT IS MADE TO CAT-
            EGORIZE COUNTS WHICH FALL 'OUT OF RANGE' (MORE + OR -5
            COUNTS FROM THE AVERAGE).

            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 '1024' CONVERSIONS
                C=THE AVERAGE READING OF THE '1024' CONVERSIONS
                D=THE HIGHEST READING OF THE '1024' CONVERSIONS
                E=NUMBER OF COUNTS IN EACH PART LOWER THAN 5 COUNTS
              F-J=NUMBER OF COUNTS IN EACH PART LOWER THAN AVERAGE.
                K=NUMBER OF COUNTS AT AVERAGE OF THE '1024'
              L-P=NUMBER OF COUNTS IN EACH PART HIGHER THAN AVERAGE.
                Q=NUMBER OF COUNTS 'OUT OF RANGE' HIGHER THAN 5 COUNTS


10. GAIN TEST
    ---------

        A.  THE GAIN TEST IS USED TO DETERMINE THE ACCURACY OF THE
            'ADF11' AT DIFFERENT GAIN SETTINGS. THE TEST REQUESTS
            16 SPECIFIED VOLTAGES (8 FOR A UNIPOLAR A/D) TO BE APPLIED
            TO THE SELECTED CHANNEL. A SERIES OF '1024' CONVERSIONS ARE
            TAKEN FOR EVERY VOLTAGE AND APPLICABLE GAIN SETTINGS AND
            THE AVERAGE IS COMPARED AGAINST THE TRUE VALUE FOR THAT
            SPECIFIED SETTING.  IF THE AVERAGE IS MORE THAN + OR -1
            COUNT FROM THE TRUE VALUE IT IS CONSIDERED IN ERROR AND
            THE CONVERSIONS 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.
```

```
  318
  319                                          -5-
```

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 SPECIFIED 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.
3. ↑P (CONTROL P)
   LOAD THE COMMAND DECODER (REFER TO SECTION 13.)

D. 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. THE TEST IS THEN LOOPED UNTIL EITHER THE CORRECT CONVERSION RESULTS ARE OBTAINED OR SW14 IS SET INHIBITING LOOPING.

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	2.50000	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		

```
   376                                           4      ------ ------ 776000 777000 777400 777600 777700
   377                                           8      ------ ------ ------ 776000 777000 777400 777600 777700
   378
   379                                                                  -6-
```

11. RECOVERY TEST

A. THE "RECOVERY TEST" IS DESIGNED TO DETERMINE THE RECOVERY CAPABILITY OF THE ANALOG COMPONETS IN THE 'ADF11'. 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 '↑A7 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.
3. ↑P (CONTROL P)
   LOAD THE COMMAND DECODER (REFER TO SECTION 13.)

D. CONSOLE SWITCH SETTINGS

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

```
                                   12.     INCREMENT MEMORY TEST
                                           ---------------------

                                           A.THE INCREMENT MEMORY TEST  IS DESIGNED TO TEST THE
                                           INCREMENT MEMORY CAPABILITIES OF THE ADF11 . THE
                                           "INC FLAG" IS TESTED FOR SUFFICIENT CORE MEMORY TO
                                           RUN THE DESIRIED TEST AND WILL PRINT " INSUFFIENT CORE MEMORY"
                                           IF THE TEST IS ATTEMPTED AND THE FLAG IS NOT SET.
                                           THE TEST SELECTS THE CHANNEL INPUT FROM THE SWITCH
                                           REGISTER BITS 0-7 AND THE GAIN REMAINS AT "X1"
                                           MEMORY INCREMENTING STARTS AT 8K (ADDRESS 020000) FOR 12
                                           BITS OR LESS AND AT 12K (ADDRESS 040000) FOR 13 BIT
                                           CONVERTERS
                                           THE INCREMENTED CORE MEMORY RANGE IS CLEARED BEFORE EACH
                                           CONVERSION AND THE STATUS WORD ADDRESS REGISTER IS READ
                                           AFTER EACH CONVERSION TO TEST THE MEMORY LOCATION INCREMENTED.
                                           IF THE CONTENTS OF THE ADDRESS IN THE (SWAR) REGISTER IS NOT
                                           INCREMENTED "CORE MEMORY LOCATION NOT INCREMENTED = XXXXXX"
                                           IS PRINTED AND THE TEST RESUMES.  IF THE LOCATION IS
                                           INCREMENTED, THE ADDRESS IS EITHER PRINTED OR DISPLAYED
```

```
                    AS DETERMINED BY (SW REG) BIT  15.

                B.  STARTING SEQUENCE
                    -----------------
                    1.  TYPE "I" FROM THE MONITOR TO RUN INCREMENT MEMORY
                    TEST FOLLOWED BY A CARRIAGE RETURN.
                    2.  A REQUEST IS THEN MADE FOR TYPE OF SYNC "INT. OR
                     EXT." TO WHICH OPERATOR MUST RESPOND FOLLOWED
                    BY A "C.R."
                    3.  SWITCH REGISTER BITS 00-07 SELECTS CHANNEL ADDRESS .
                    4.  SWITCH REGISTER BIT 15=1 THE COMPOSITE ADDRESS
                    (SWAR) IS PRINTED (9/LINE).  SWITCH REGISTER BIT 15=0
                    THE (SWAR) IS DISPLAYED THROUGH R0.

                C.  CONTROL SWITCHES (TELETYPE)
                    ---------------------------
                    1.  ↑A (CONTROL A) TYPING A CONTROL A WILL ENABLE
                    FOR THE "SYNC" TO BE CHANGED FROM INT. OR EXT.
                    2.↑C (CONTROL C) TYPING A CONTROL C WILL ENABLE THE PROGR
                    AM TO RETURN TO THE MONITOR.
                    3.  ↑P (CONTROL P) LOAD THE COMMAND DECODER (REFER TO
                    SECTION 13).
                                       -8-
```

```
                        13. COMMAND DECODER
                            ---------------

                                A.  THE 'COMMAND DECODER' IS USED TO ENABLE THE USER TO EXIT
                                    ANY SEQUENCE OF TESTING.  HE CAN THEN CHANGE ANY ONE OR
                                    MORE OF THE PARAMETERS FOR THAT TEST AND CONTINUE THE
                                    TESTING SEQUENCE WITHOUT CHANGING ALL PARAMETERS.

                                B.  OPERATE INSTRUCTIONS
                                    --------------------

                                    1.  TYPE '↑P' TO ENTER THE COMMAND DECODER.  THE PROGRAM WILL
                                        RESPOND VIA TYPING AN ASTRIC (*) TO INDICATE READY.
                                        THE USER CAN THEN CHANGE; CHANNELS(S), GAINS(S),COUNT
                                        SPREAD OR SYNC AT HIS DESCRETION.

                                C.  COMMANDS
                                    --------

                                        TYPE            PARAMETER AFFECTED
                                        ----            ------------------

                                        C  (CR)         CHANNEL(S)
                                        G  (CR)         GAIN(S)
                                        CS (CR)         COUNT SPREAD
                                        S  (CR)         SYNC
                                        ST (CR)         EXIT AND CONTINUE TESTING

                        14. LISTING
                            -------

                                                        -9-
                                %
                                .TITLE ADF11A  PART II, ANALOG DIAGNOSTIC TEST
                                .ABS
                        ;MAINDEC-11-DZADH-A-D
                        ;COPYRIGHT 23-JULY-75
                        ;DIGITAL EQUIPMENT CORP. MAYNARD MASS. 01754
                        ;PROGRAMMERS: EARL L. BOUSE/R.BALDWIN

                        ;SWITCH REGISTER DEFINITIONS AND FUNCTIONS:

        100000          SW15=100000
        040000          SW14=40000
        020000          SW13=20000
        010000          SW12=10000
        004000          SW11=4000
        002000          SW10=2000
        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=000
        020000                  G2=20000
        040000                  G4=40000
        060000                  G8=60000
                                ;INSTRUCTIONS DEFINITIONS

        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
```

```
    560
    561                                              ;LOAD TRAP CATCHER INTO LOC'S 0-1000
    562               000000                                 .=0
    563     000000    000002                                 .+2
    564     000002    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    565     000004    000006                                 .+2
    566     000006    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    567     000010    000012                                 .+2
    568     000012    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    569     000014    000016                                 .+2
    570     000016    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    571     000020    000022                                 .+2
    572     000022    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    573     000024    000026                                 .+2
    574     000026    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    575     000030    000032                                 .+2
    576     000032    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    577     000034    000036                                 .+2
    578     000036    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    579     000040    000042                                 .+2
    580     000042    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    581     000044    000046                                 .+2
    582     000046    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    583     000050    000052                                 .+2
    584     000052    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    585     000054    000056                                 .+2
    586     000056    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    587     000060    000062                                 .+2
    588     000062    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    589     000064    000066                                 .+2
    590     000066    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    591     000070    000072                                 .+2
    592     000072    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    593     000074    000076                                 .+2
    594     000076    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    595     000100    000102                                 .+2
    596     000102    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    597     000104    000106                                 .+2
    598     000106    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    599     000110    000112                                 .+2
    600     000112    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    601     000114    000116                                 .+2
    602     000116    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    603     000120    000122                                 .+2
    604     000122    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    605     000124    000126                                 .+2
    606     000126    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    607     000130    000132                                 .+2
    608     000132    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    609     000134    000136                                 .+2
    610     000136    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    611     000140    000142                                 .+2
    612     000142    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    613     000144    000146                                 .+2
    614     000146    000004                                 4                          ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    615     000150    000152                                 .+2
```

```
616  000152  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
617  000154  000156                          .+2
618  000156  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
619  000160  000162                          .+2
620  000162  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
621  000164  000166                          .+2
622  000166  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
623  000170  000172                          .+2
624  000172  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
625  000174  000176                          .+2
626  000176  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
627  000200  000202                          .+2
628  000202  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
629  000204  000206                          .+2
630  000206  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
631  000210  000212                          .+2
632  000212  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
633  000214  000216                          .+2
634  000216  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
635  000220  000222                          .+2
636  000222  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
637  000224  000226                          .+2
638  000226  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
639  000230  000232                          .+2
640  000232  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
641  000234  000236                          .+2
642  000236  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
643  000240  000242                          .+2
644  000242  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
645  000244  000246                          .+2
646  000246  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
647  000250  000252                          .+2
648  000252  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
649  000254  000256                          .+2
650  000256  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
651  000260  000262                          .+2
652  000262  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
653  000264  000266                          .+2
654  000266  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
655  000270  000272                          .+2
656  000272  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
657  000274  000276                          .+2
658  000276  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
659  000300  000302                          .+2
660  000302  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
661  000304  000306                          .+2
662  000306  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
663  000310  000312                          .+2
664  000312  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
665  000314  000316                          .+2
666  000316  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
667  000320  000322                          .+2
668  000322  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
669  000324  000326                          .+2
670  000326  000004                          4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
671  000330  000332                          .+2
```

```
   672  000332  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   673  000334  000336                                  .+2
   674  000336  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   675  000340  000342                                  .+2
   676  000342  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   677  000344  000346                                  .+2
   678  000346  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   679  000350  000352                                  .+2
   680  000352  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   681  000354  000356                                  .+2
   682  000356  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   683  000360  000362                                  .+2
   684  000362  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   685  000364  000366                                  .+2
   686  000366  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   687  000370  000372                                  .+2
   688  000372  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   689  000374  000376                                  .+2
   690  000376  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   691  000400  000402                                  .+2
   692  000402  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   693  000404  000406                                  .+2
   694  000406  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   695  000410  000412                                  .+2
   696  000412  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   697  000414  000416                                  .+2
   698  000416  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   699  000420  000422                                  .+2
   700  000422  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   701  000424  000426                                  .+2
   702  000426  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   703  000430  000432                                  .+2
   704  000432  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   705  000434  000436                                  .+2
   706  000436  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   707  000440  000442                                  .+2
   708  000442  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   709  000444  000446                                  .+2
   710  000446  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   711  000450  000452                                  .+2
   712  000452  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   713  000454  000456                                  .+2
   714  000456  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   715  000460  000462                                  .+2
   716  000462  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   717  000464  000466                                  .+2
   718  000466  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   719  000470  000472                                  .+2
   720  000472  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   721  000474  000476                                  .+2
   722  000476  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   723  000500  000502                                  .+2
   724  000502  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   725  000504  000506                                  .+2
   726  000506  000004                                  4                                  ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   727  000510  000512                                  .+2
```

```
    728  000512  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    729  000514  000516                  .+2
    730  000516  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    731  000520  000522                  .+2
    732  000522  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    733  000524  000526                  .+2
    734  000526  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    735  000530  000532                  .+2
    736  000532  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    737  000534  000536                  .+2
    738  000536  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    739  000540  000542                  .+2
    740  000542  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    741  000544  000546                  .+2
    742  000546  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    743  000550  000552                  .+2
    744  000552  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    745  000554  000556                  .+2
    746  000556  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    747  000560  000562                  .+2
    748  000562  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    749  000564  000566                  .+2
    750  000566  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    751  000570  000572                  .+2
    752  000572  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    753  000574  000576                  .+2
    754  000576  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    755  000600  000602                  .+2
    756  000602  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    757  000604  000606                  .+2
    758  000606  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    759  000610  000612                  .+2
    760  000612  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    761  000614  000616                  .+2
    762  000616  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    763  000620  000622                  .+2
    764  000622  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    765  000624  000626                  .+2
    766  000626  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    767  000630  000632                  .+2
    768  000632  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    769  000634  000636                  .+2
    770  000636  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    771  000640  000642                  .+2
    772  000642  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    773  000644  000646                  .+2
    774  000646  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    775  000650  000652                  .+2
    776  000652  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    777  000654  000656                  .+2
    778  000656  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    779  000660  000662                  .+2
    780  000662  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    781  000664  000666                  .+2
    782  000666  000004                  4                               ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
    783  000670  000672                  .+2
```

```
   784 000672 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   785 000674 000676                         .+2
   786 000676 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   787 000700 000702                         .+2
   788 000702 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   789 000704 000706                         .+2
   790 000706 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   791 000710 000712                         .+2
   792 000712 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   793 000714 000716                         .+2
   794 000716 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   795 000720 000722                         .+2
   796 000722 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   797 000724 000726                         .+2
   798 000726 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   799 000730 000732                         .+2
   800 000732 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   801 000734 000736                         .+2
   802 000736 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   803 000740 000742                         .+2
   804 000742 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   805 000744 000746                         .+2
   806 000746 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   807 000750 000752                         .+2
   808 000752 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   809 000754 000756                         .+2
   810 000756 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   811 000760 000762                         .+2
   812 000762 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   813 000764 000766                         .+2
   814 000766 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   815 000770 000772                         .+2
   816 000772 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   817 000774 000776                         .+2
   818 000776 000004                         4                                 ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
   819
   820        000020                         .=20
   821 000020 006576                         ERTRAP                            ;ERROR TRAP REPORTER ROUTINE
   822 000022 000340                         340
   823 000024 006232                         PWRFAL                            ;POWER FAIL HANDLER
   824 000026 000340                         340
   825        000060                         .=60
   826 000060 005410                         XTTYIN                            ;TELEPRINTER KEYBOARD ROUTINE
   827 000062 000340                         340
   828        000030                         .=30
   829 000030 007166                         EMTSRV                            ;EMT TRAP, EMT DISPATCH SERVICE
   830 000032 000340                         340
   831        000174                         .=174
   832 000174 000167 001042                  JMP     INIT                      ;INITIALIZATION ADDRESS
   833 000200 000167 001362                  JMP     MONITR                    ;PROGRAM 'RESTART' ADDRESS
```

```
                                  ;TRAP EQUIVALENCE TABLE:

                  104000          PRINT=EMT                 ;MESSAGE PRINTER ROUTINE
                  104001          DECOCT=EMT+1              ;DECIMAL TO OCTAL CONVERSIN ROUTINE
                  104002          RDMEM=EMT+2               ;SUBROUTINE TO READ CATEGORIZE INC. MEM. VALUES
                  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          GETCHA=EMT+17             ;SUBROUTINE TO DECODE A CH(S) INPUT FROM TTY.
                  104020          SAVREG=EMT+20             ;SUBROUTINE TO SAVE REG.'S ON THE STACK
                  104021          GETREG=EMT+21             ;SUBROUTINE TO GET REG.'S FROM THE STACK

                                  ;REGISTER ADDRESSES
                  001200                  .=1200
         001200   177776          PSW:    177776            ;ADDRESS OF PROCESSOR STATUS REG.
         001202   177560          TKS:    177560            ;ADDRESS OF KEYBOARD STATUS REG.
         001204   177562          TKB:    177562            ;   "     "     "      BUFFER  "
         001206   177564          TPS:    177564            ;   "     "  PRINTER STATUS REG.
         001210   177566          TPB:    177566            ;   "     "  PRINTER BUFFER REG.
         001212   177570          SWR:    177570            ;   "     "  SWITCH REG.
         001214   177571          SWRO:   177571            ;   "     "    "     "   HIGH BYTE
         001216   164006          ADCR:   164006            ;   "     "  A/D CONTROL REG.
         001220   164010          ADCSR:  164010            ;   "     "  A/D CONTROL & STATUS REG.
         001222   164000          ADSWR:  164000            ;   "     "  A/D STATUS WORD REG
         001224   164004          ADWCR:  164004            ;   "     "  A/D WORD COUNT REG.
         001226   164012          ADDBR:  164012            ;   "     "  A/D DATA BUFFER REG.
         001230   164002          ADWRA:  164002            ;   "     "  A/D WORD REG 'A'
         001232   164014          ADWRB:  164014            ;   "     "  A/D   "    "  'B'
         001234   164016          ADAOR:  164016            ;ADDRESS OF A/D OFFSET REG.
         001236   000274          ADINT:  0274              ;ADDRESS OF THE A/D INTERRUPT VECTOR
         001240   000276          ADLVL:  0276              ;ADDRESS OF THE A/D INTERRUPT LEVEL
```

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

001242  016706  011530          INIT:   MOV     STACK,SP        ;INIT STACK POINTER=1000
001246  012777  000340  177724          MOV     #340,@PSW
001254  104000                          PRINT                   ;CALL MESSAGE PRINTER VIA 'EMT'
001256  011001                          TITLE                   ;TYPE PROGRAM HEADER.
001260  005067  011504          INIT1:  CLR     ADSIGN          ;UNIPOLAR=0,BIPOLAR=1
001264  104000                          PRINT
001266  011105                          MES2                    ;REQUEST THE A/D LENGTH
001270  104001                          DECOCT                  ;CONVERT A/D LENGTH TO OCTAL
001272  012701  003776                  MOV     #3776,R1        ;INIT AS 'INC MEM' OFFSET
001276  012702  001000                  MOV     #1000,R2        ;= TO +5V VALUE FOR 10 BITS
001302  162767  000012  004554          SUB     #12,BCDTAB      ;A/D LENGTH = TO 10 BITS?
001310  001414                          BEQ     LDSIZE          ;YES, EXIT
001312  012703  000005                  MOV     #5,R3           ;NO, TEST UP TO 15 BITS
001316  006301                  SIZE:   ASL     R1              ;BUMP MEM. OFFSET
001320  006302                          ASL     R2              ;ALSO A/D SIZE
001322  005367  004536                  DEC     BCDTAB          ;DECREMENT COUNT
001326  001405                          BEQ     LDSIZE          ;EXIT IF DONE
001330  005303                          DEC     R3
001332  100371                          BPL     SIZE            ;BRANCH UNTIL 15 IS REACHED
001334  104000                          PRINT                   ;ILLEGAL ENTRY
001336  011463                          QMARK                   ;PRINT '?'
001340  000747                          BR      INIT1           ;RETRY
001342  012700  013172          LDSIZE: MOV     #POS500,R0      ;LOAD +5V VALUE
001346  010203                          MOV     R2,R3           ;SAVE AD SIZE
001350  010320                  LDPOS:  MOV     R3,(R0)+
001352  006203                          ASR     R3
001354  022700  013204                  CMP     #NEG500,R0      ;LOADED ALL POS. VAUES?
001360  001373                          BNE     LDPOS           ;BRANCH IF NO
001362  010203                          MOV     R2,R3           ;RESET +5V VALUE
001364  005403                          NEG     R3
001366  010320                  LDNEG:  MOV     R3,(R0)+
001370  006203                          ASR     R3
001372  022700  013216                  CMP     #NEG312+2,R0    ;LOADED ALL NEG. VALUES?
001376  001373                          BNE     LDNEG           ;BRANCH IF NO.
001400  005767  011364                  TST     ADSIGN          ;TEST FOR SIGN BIT
001404  001401                          BEQ     CORSIZ          ;BRANCH IF NOT SET
001406  006301                          ASL     R1              ;OTHERWISE ADD 1 BIT TO CONVERTER LENGTH.
001410  052701  000776          CORSIZ: BIS     #776,R1         ;SET ALL POSSIBLE SHIFTED BITS
001414  012737  001476  000004          MOV     #INITA,@#4      ;INITIAL THE TIME OUT ADDRESS
001422  012737  000340  000006          MOV     #340,@6
001430  005067  011350                  CLR     INCFLG          ;CLR INCREMENT MEMORY FLAG
001434  012767  020000  011346          MOV     #20000,OFFSET   ;INC MEM.STARTS @20000 FOR <13BITS<<<<
001442  032701  020000                  BIT     #20000,R1       ;TEST FOR 13 BIT CONVERTER<<<<
001446  001405                          BEQ     .+14            ;BRANCH IF NOT<<<<
001450  062701  020000                  ADD     #20000,R1       ;ADD AN ADDITIONAL 4K OFFSET FOR 13<<<<
001454  012767  040000  011326          MOV     #40000,OFFSET   ;13 BITS INC MEM STARTS AT <<<<
001462  062701  020000                  ADD     #20000,R1       ;ADD 4K OFFSET TO A/D LENGTH<<<<
001466  005711                          TST     @R1             ;TEST IF MEMORY IS AVAILABLE<<<<
001470  012767  000377  011306          MOV     #377,INCFLG     ;YES SETINGS FOR INCREMENT MEMORY
```

```
001476  012737  000006  000004  INITA:  MOV     #6,@#4          ;RESET TIME OUT ADDRESS
001504  005067  176276                  CLR     6               ;TO HALT ON TIMEOUT
001510  010167  011272                  MOV     R1,MEMSIZ       ;SAVE MEMORY SIZE
001514  006302                          ASL     R2              ;SET UP OFFSET FOR AVERAGING ROUTINE
001516  010267  011272                  MOV     R2,ADSIZE       ;SAVE IT
001522  012701  176000                  MOV     #176000,R1      ;BASE VAL OF AD EXT OF SIGN
001526  012700  002000                  MOV     #2000,R0        ;BASE VAL OF A/D SIZE
001532  030067  011256          1$:     BIT     R0,ADSIZE       ;BIILD SIGN EXT ACCOR TO AD-SIZE
001536  001006                          BNE     2$
001540  006301                          ASL     R1              ;SHORTEN AD SIGN EXT
001542  006300                          ASL     R0              ;SCALE FOR AD SIZE
001544  103372                          BCC     1$              ;IF CARRY SETS WE ARE OUT OF RANGE
001546  104000  011463                  PRINT,QMARK
001552  000642                          BR      INIT1
001554  010167  011212          2$:     MOV     R1,SIGEXT       ;SAVE SIGN EXT FOR MEM.INC
001560  104000                  INITB:  PRINT
001562  011264                          MES4                    ;PRINT THE TEST CALL LETTERS
001564  000407                          BR      INIT2           ;GO AND AWAIT COMMAND


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

001566  000005                  MONITR: RESET                   ;INITIALIZE ON ENTRY
001570  016706  011202                  MOV     STACK,SP        ;RESET STACK POINTER
001574  004767  007150                  JSR     PC,CLRINT       ;CLR A/D INTERRUPT VECTOR
001600  104000                          PRINT                   ;CALL MESSAGE PRINTER
001602  011441                          CNTRLC                  ;TYPE '↑C'

001604  012767  001560  011166  INIT2:  MOV     #INITB,AVECTR   ;SET UP '↑A' VECTOR ADDRESS.
001612  012767  001722  011162          MOV     #INIT3,PVECTR   ;SET UP '↑P' VECTOR ADDRESS
001620  104000                          PRINT
001622  011460                          DOT                     ;PRINT '.' TO INDICATE MONITOR READY
001624  104011                          TTYIN                   ;WAIT FOR TTY ENTRY
001626  122767  000103  004034          CMPB    #103,INBUF      ;TEST FOR 'C'
001634  001002                          BNE     .+6             ;NOT 'C'
001636  000167  000066                  JMP     CALBRT          ;YES, RUN 'CALIBRATION' TEST
001642  122767  000122  004020          CMPB    #122,INBUF      ;TEST FOR 'R'
001650  001002                          BNE     .+6             ;NOT 'N'
001652  000167  000350                  JMP     REPTST          ;YES, RUN 'REPEATIBILITY' TEST
001656  122767  000107  004004          CMPB    #107,INBUF      ;TEST FOR 'G'
001664  001002                          BNE     .+6             ;NOT 'G'
001666  000167  001014                  JMP     GAIN            ;YES, RUN 'GAIN' TEST
001672  022767  000105  003770          CMP     #105,INBUF      ;TEST FOR 'E'
001700  001002                          BNE     .+6             ;NOT 'E'
001702  000167  002370                  JMP     RECVRY          ;YES, RUN RECOVERY TEST
001706  022767  000111  003754          CMP     #111,INBUF              ;TEST FOR 'I'
001714  001002                          BNE     .+6             ;NOT I
001716  000167  002546                  JMP     INCTST          ; YES RUN INC MEM.
001722  104000                  INIT3:  PRINT                   ;ILLEGAL ENTRY
001724  011463                          QMARK                   ;TYPE '?'
001726  000726                          BR      INIT2           ;WAIT AGAIN
```

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

                                                ;ROUTINE REQUESTS THE TYPE OF 'SYNC' TO BE USED ('I' INTERNAL OR 'E' EXTERNAL).
                                                ;THE PROGRAM THEN TAKES CONTINUOUS CONVERSIONS (SEQ. DMA MODE) USING DATA
                                                ;SW'S 0-8 TO SELECT THE CH.  SW'S 13&14 TO SELECT GAIN AND EITHER
                                                ;SW'S 9-11 TO SELECT DELAY OR SW '15' TO PRINT THE CONVERSION VALUE.

001730  012767  001756  011042  CALBRT: MOV     #CALBT1,AVECTR  ;SET UP '↑A' RESTART ADDRESS
001736  012767  002004  011036          MOV     #CALBT2,PVECTR  ;SET UP '↑P' START ADDRESS.
001744  104000                          PRINT
001746  011513                          MES7                    ;TYPE TEST HEADER
001750  104000                          PRINT
001752  011567                          MES10                   ;TEXT 'SYNC I OR E'
001754  000402                          BR      CALB1A          ;WAIT FOR INPUT
001756  104000                  CALBT1: PRINT
001760  012441                          MES39                   ;TEST 'SYNC?'
001762  104011                  CALB1A: TTYIN                   ;WAIT FOR INPUT.
001764  016767  003700  011024          MOV     INBUF,PROC      ;SAVE IT IN TEMP STORAGE
001772  012767  000011  011050          MOV     #11,KSTOR3      ;PRINT '9' CONVERSIONS/LINE
002000  104000                          PRINT
002002  011453                          CALF
002004  012767  000001  011012  CALBT2: MOV     #1,ICOUNT       ;SETUP TO PRINT '1' VALUE
002012  017767  177174  011016          MOV     @SWR,FINAL      ;GET CH. FROM SW. REG.
002020  042767  177000  011010          BIC     #177000,FINAL   ;CLR UNWANTED BITS
002026  017767  177160  011000          MOV     @SWR,INITAL     ;ALSO GET AS INITIAL CH. & GAIN
002034  042767  107000  010772          BIC     #107000,INITAL  ;CLR UNWANTED BITS
002042  052767  110000  010764          BIS     #110000,INITAL  ;SELECT SEQ. DMA, INITAL
002050  017767  177136  010766          MOV     @SWR,KSTOR1     ;GET DELAY BITS 9-11
002056  000367  010762                  SWAB    KSTOR1
002062  006267  010756                  ASR     KSTOR1
002066  042767  177770  010750          BIC     #177770,KSTOR1  ;DELAY NOW SET
002074  017767  177112  010744          MOV     @SWR,KSTOR2     ;SAVE ORIGINAL SWITCH SETTING.
002102  022767  000105  010706          CMP     #105,PROC       ;TEST SYNC SELECT
002110  001003                          BNE     CALB2A          ;BRANCH IF NOT 'E'
002112  052767  004000  010714          BIS     #4000,INITAL    ;SET 'EXT' SYNC ENABLE
002120  012777  177777  177076  CALB2A: MOV     #-1,@ADWCR      ;SET UP FOR '1' CONVERSION
002126  104016                          TSTTKS
002130  004767  003740                  JSR     PC,ADCNVT       ;TAKE AND STORE THE CONVERSIONS
002134  005777  177052                  TST     @SWR            ;TEST FOR SW15 TO PRINT
002140  100015                          BPL     CALB2B          ;BRANCH IF NOT SET.
002142  012767  013434  004766          MOV     #ADBUFF,AVGTAB  ;SET UP TO PRINT VALUE
002150  104014                          PRTAVG                  ;PRINT IT
002152  005367  010672                  DEC     KSTOR3
002156  001015                          BNE     CALBT4
002160  012767  000011  010662          MOV     #11,KSTOR3
002166  104000                          PRINT
002170  011453                          CALF
002172  000407                          BR      CALBT4          ;TEST FOR LOOP
```

```
002174  016700  011234          CALB2B: MOV     ADBUFF,R0       ;SET UP A/D BUFFER.
002200  016701  010640                  MOV     KSTOR1,R1       ;SET UP DELAY (RESET COUNT)
002204  000005                  CALBT3: RESET
002206  005301                          DEC     R1              ;DECREMENT DELAY
002210  100375                          BPL     CALBT3
002212  104016                  CALBT4: TSTTKS                  ;TEST FOR KEYBOARD INTERRUPT
002214  026777  010626  176770          CMP     KSTOR2,@SWR     ;TEST IF SWITCH REGISTER HAS CHANGED
002222  001736                          BEQ     CALB2A          ;BRANCH AND TAKE NEXT CONVERSION
002224  000667                          BR      CALBT2          ;YES, COMPUTE NEW INPUT
```

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

                                ;THIS ROUTINE TO DESIGNED TO SHOW REPEATIBILITY BY TAKING A SERIES OF
                                ;'1024'  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.  IF THERE IS SUFFICIENT MEMORY AVAILABLE THE USER
                                ;IS GIVEN THE OPTION OF USING THE INCREMENT MEMORY MODE.  A CONTINUOUS
                                ;SERIES OF CONVERSIONS ARE THEN TAKEN AND COMPARED AGAINST THE INPUT
                                ;COUNT SPREAD.  IF ALL '1024' 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).

002226  012767  002240  010544  REPTST: MOV     #REPT1,AVECTR    ;SET UP CNTR 'A' VECTOR ADDRESS
002234  104000                          PRINT
002236  011540                          MES13                    ;TEXT 'REPEATIBILITY TEST'
002240  005067  010546          REPT1:  CLR     SOFLAG           ;CLR SOFTWARE FLAG
002244  005067  010564                  CLR     INITAL           ;CLR CH. STORAGE REG.
002250  005067  010620                  CLR     MESPRT           ;CLR PRINT SW.
002254  012767  002350  010520          MOV     #REPT2,PVECTR    ;SET UP '↑P' VECTOR ADDRESS
002262  005767  010516                  TST     INCFLG           ;TEST IF FLAG IS SET
002266  001420                          BEQ     REPT1A           ;BRANCH IF NO
002270  104000                          PRINT                    ;OTHERWISE GIVE INC. MEM. OPTION
002272  011466                          MES6                     ;TEXT 'INC MEM. (Y OR N)?'
002274  104011                          TTYIN                    ;WAIT FOR REPLY
002276  022767  000131  003364          CMP     #131,INBUF       ;WAS 'Y' TYPED?
002304  001011                          BNE     REPT1A           ;BRANCH IF NO.
002306  012767  000377  010476          MOV     #377,SOFLAG      ;YES, SET SOFTWARE FLAG.
002314  052767  002000  010512          BIS     #2000,INITAL     ;SET INCREMENT MEMORY BIT
002322  016777  010462  176704          MOV     OFFSET,@ADADR    ;LOAD THE OFFSET REG.+++++++++++++++
002330  104017                  REPT1A: GETCHA                   ;REQUEST & STORE CH(S)
002332  104003                          GAININ                   ;SET UP GAIN
002334  104000                          PRINT
002336  011701                          MES16                    ;TEXT 'COUNT SPREAD ?'
002340  104001                          DECOCT                   ;DECODE TO OCTAL
002342  016767  003516  010500          MOV     BCDTAB,KSTOR3    ;SAVE IT
002350  016767  010462  010466  REPT2:  MOV     FINAL,KSTOR1     ;SAVE STARTING CH.
002356  052767  110000  010450          BIS     #110000,INITAL   ;SELECT; SEQ. DMA
002364  016767  010454  010444  REPT2A: MOV     KSTOR1,FINAL     ;RESET FINAL ADDRESS
002372  042767  001777  010434          BIC     #1777,INITAL
002400  056767  010432  010426          BIS     FINAL,INITAL     ;RESET THE INIAL CH.
002406  016767  010432  010436          MOV     KSTOR1,KSTOR4    ;SAVE STARTING CH,
002414  004767  002706          REPT3:  JSR     PC,CLACOR        ;CLR MEM FOR 'INC'
002420  012777  176000  176576          MOV     #-2000,@ADWCR    ;SET FOR '1024' CONVERSIONS
002426  104016                          TSTTKS
002430  004767  003440                  JSR     PC,ADCNVT        ;TAKE THE CONVERSIONS
002434  005767  010352                  TST     SOFLAG           ;INC. MEM?
```

```
 002440  001402                          BEQ     REPT3A          ;NO, EBA (EARL BOUSE ARITHMETIC)
 002442  104002                          RDMEM                   ;YES, READ MEMORY & COLLECT VALUES
 002444  000402                          BR      REPT3B
 002446  104004                 REPT3A:  CMPUTE                  ;AVERAGE & COMPUTE DISTRIBUTION
 002450  104005                          CATORIZ
 002452  032777  010000  176532 REPT3B:  BIT     #SW12,@SWR      ;TEST DATA SW12
 002460  001025                          BNE     REPT4           ;IF SET, FORCE TYPE OUT
 002462  032777  020000  176522 TSTCT4:  BIT     #SW13,@SWR      ;TEST FOR INHIBIT TYPEOUT
 002470  001073                          BNE     REPT7           ;BRANCH IF SW SET
 002472  012700  000001                  MOV     #1,R0
 002476  012701  013162                  MOV     #XSPRD1,R1
 002502  016702  010342                  MOV     KSTOR3,R2
 002506  020002                 TSTCNT:  CMP     R0,R2
 002510  001406                          BEQ     CHKCNT
 002512  005200                          INC     R0
 002514  005721                          TST     (R1)+           ;UPDATE COUNT ADDRESS
 002516  020127  013172                  CMP     R1,#XSPRD4+2    ;CHECKED '1-4'?
 002522  001371                          BNE     TSTCNT          ;NO
 002524  000403                          BR      REPT4           ;ILLEGAL ENTRY, TYPE OUT COUNTS
 002526  022711  002000         CHKCNT:  CMP     #2000,(R1)      ;ARE ALL COUNTS IN COUNT SPREAD?
 002532  001452                          BEQ     REPT7           ;YES, EXIT
 002534  104000                 REPT4:   PRINT
 002536  011453                          CRLF
 002540  005767  010330                  TST     MESPRT          ;TEST IF HEADER HAS BEEN TYPED
 002544  001002                          BNE     REPT5           ;BRANCH IF YES
 002546  104000                          PRINT
 002550  011732                          MES19                   ;TEXT 'CH. HIGH AVG. LOW'
```

```
002552  104000                          REPT5:  PRINT
002554  011453                                  CRLF                    ;CARRIAGE RETURN, LINE FEED
002556  016702  010270                          MOV     KSTOR4,R2       ;MOV. CH.
002562  104006                                  BINDEC                  ;CONVERT TO DECIMAL AND PRINT
002564  104007                                  SPACE
002566  104010                                  PRTOCT                  ;PRINT LOW VALUE
002570  013100                                  LOW
002572  104007                                  SPACE
002574  104010                                  PRTOCT                  ;PRINT AVERAGE VALUE
002576  013114                                  AVRAGE
002600  104007                                  SPACE
002602  104010                                  PRTOCT                  ;PRINT HIGH VALUE
002604  013076                                  HIGH
002606  005767  010262                          TST     MESPRT
002612  001002                                  BNE     REPT6
002614  104000                                  PRINT
002616  011764                                  MES20                   ;PRINT 'COUNT SPREAD' HEADER
002620  052767  000007  010246  REPT6:  BIS     #7,MESPRT       ;INHIBIT OTHER HEADERS
002626  022767  002000  010310          CMP     #2000,AVGCNT    ;TEST IF ALL COUNTS WERE AT AVG.
002634  001411                                  BEQ     REPT7           ;BRANCH TO NEXT CH. IF YES.
002636  104000                                  PRINT
002640  011453                                  CRLF
002642  012704  013130                          MOV     #ORLOW,R4
002646  012402                          REPT6A: MOV     (R4)+,R2
002650  104006                                  BINDEC                  ;TYPE OUT COUNT SPREAD
002652  022704  013162                          CMP     #XSPRD1,R4      ;TEST FOR DONE
002656  001373                                  BNE     REPT6A          ;BRANCH IF NO AND TYPE NEXT COUNT
002660  005267  010150                  REPT7:  INC     INITAL
002664  005267  010146                          INC     FINAL
002670  005267  010156                          INC     KSTOR4          ;INCREMENT 'CH.'
002674  026767  010142  010150          CMP     FINAL2,KSTOR4   ;TESTED ALL CH.(S)?
002702  002244                                  BGE     REPT3           ;BRANCH IF NO AND TEST NEXT CH.
002704  000627                                  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 '1024' 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.

002706  012767  002720  010064  GAIN:   MOV     #GT,AVECTR      ;SET UP '↑A' RETURN ADDRESS
002714  104000                          PRINT
002716  011123                          MES3                    ;TEXT GAIN ACCURACY TEST
002720  005067  010110          GT:     CLR     INITAL
002724  012767  002734  010050          MOV     #GTO+2,PVECTR   ;SET UP '↑P' VECTOR ADDRESS
002732  104017                  GTO:    GETCHA                  ;REQUEST CH.
002734  052767  110000  010072          BIS     #110000,INITAL  ;SEQ., DMA, INITAL
002742  005067  010126                  CLR     MESPRT          ;CLR PRINT INHIBIT SWITCH

                                ;TEST +5.0V X G1

002746  104000                  GT1:    PRINT
002750  012115                          MES23                   ;TEXT '+5.00V'
002752  104012                          WAITGN                  ;CALL THE GAIN 'WAIT' HANDLER
002754  000000                          G1                      ;GAIN X1
002756  013172                          POS500
002760  013216                          GP50X1                  ;SAVE VALUE
002762  005767  010002                  TST     ADSIGN
002766  001406                          BEQ     GT2             ;BRANCH TO NEXT TEST IF UNIPOLAR.

                                ;TEST -5.0V X G1

002770  104000                          PRINT
002772  012124                          MES24                   ;TEXT 'SWITCH VOLTAGE NEG.'
002774  104012                          WAITGN
002776  000000                          G1                      ;GAIN X1
003000  013204                          NEG500                  ;SHOULD=-5.0V
003002  013266                          GM50X1                  ;SAVE VALUE

                                ;TEST +2.5V X G1

003004  104000                  GT2:    PRINT
003006  012141                          MES25                   ;TEXT '+2.5V'
003010  104012                          WAITGN
003012  000000                          G1                      ;GAIN X1
003014  013174                          POS250
003016  013220                          GP25X1                  ;SAVE VALUE

```

```
                                ;TEXT +2.5V X G2 (5.0V)
003020  104013                          TAKEGN
003022  020000                          G2                      ;GAIN X2
003024  013172                          POS500                  ;SHOULD=+5.0V
003026  013230                          GP25X2                  ;SAVE VALUE
003030  005767  007734                  TST     ADSIGN
003034  001412                          BEQ     GT3             ;BRANCH IF UNIPOLAR


                                ;TEST -2.5V X G1
003036  104000                          PRINT
003040  012124                          MES24                   ;TEXT SWITCH VOLTAGE NEG.
003042  104012                          WAITGN
003044  000000                          G1                      ;GAIN X1
003046  013206                          NEG250
003050  013270                          GM25X1                  ;SAVE VALUE


                                ;TEST -2.5V X G2
003052  104013                          TAKEGN
003054  020000                          G2                      ;GAIN X2
003056  013204                          NEG500
003060  013300                          GM25X2                  ;SAVE VALUE


                                ;TEST +1.25V X G1
003062  104000                  GT3:    PRINT
003064  012150                          MES26                   ;TEXT '+1.25V'
003066  104012                          WAITGN
003070  000000                          G1                      ;GAIN X1
003072  013176                          POS125
003074  013222                          GP12X1                  ;SAVE VALUE


                                ;TEST '+1.25V X G2
003076  104013                          TAKEGN
003100  020000                          G2                      ;GAIN X2
003102  013174                          POS250                  ;=TO +2.5V
003104  013232                          GP12X2                  ;SAVE VALUE


                                ;TEST '+1.25 X G4
003106  104013                          TAKEGN
003110  040000                          G4                      ;GAIN X4
003112  013172                          POS500
003114  013242                          GP12X4                  ;SAVE VALUE
003116  005767  007646                  TST     ADSIGN
003122  001416                          BEQ     GT4             ;BRANCH IF UNIPOLAR


                                ;TEST '-1.25V X G1
003124  104000                          PRINT
003126  012124                          MES24                   ;TEXT 'SWITCH VOLTAGE NEG.'
003130  104012                          WAITGN
003132  000000                          G1                      ;GAIN X1
003134  013210                          NEG125
003136  013272                          GM12X1                  ;SAVE VALUE

```

```

003140  104013                          ;TEST -1.25V X G2
                                                 TAKEGN
003142  020000                                   G2                         ;GAIN X2
003144  013206                                   NEG250
003146  013302                                   GM12X2                     ;SAVE VALUE


                                        ;TEST -1.25V X G4
003150  104013                                   TAKEGN
003152  040000                                   G4                         ;GAIN X4
003154  013204                                   NEG500                     ;SHOULD = 5.0V
003156  013312                                   GM12X4                     ;SAVE VALUE


                                        ;TEST +0.625V X G1
003160  104000                          GT4:     PRINT
003162  012157                                   MES27                      ;TEXT '+0.625V'
003164  104012                                   WAITGN
003166  000000                                   G1
003170  013200                                   POS625                     ;SHOULD = +0.625V
003172  013224                                   GP62X1                     ;SAVE VALUE


                                        ;TEST +0.625V X G2
003174  104013                                   TAKEGN
003176  020000                                   G2                         ;GAIN X2
003200  013176                                   POS125                     ;SHOULD = +1.25V
003202  013234                                   GP62X2                     ;SAVE IT


                                        ;TEST +0.625V X G4
003204  104013                                   TAKEGN
003206  040000                                   G4                         ;GAIN X4
003210  013174                                   POS250                     ;SHOULD = +2.5V
003212  013244                                   GP62X4                     ;SAVE IT


                                        ;TEST +0.625V X G8
003214  104013                                   TAKEGN
003216  060000                                   G8                         ;GAIN X8
003220  013172                                   POS500                     ;SHOULD = +5.00V
003222  013254                                   GP62X8                     ;SAVE IT
003224  005767  007540                           TST      ADSIGN
003230  001422                                   BEQ      GT5               ;BRANCH IF UNIPOLAR


                                        ;TEST -0.625V X G1
003232  104000                                   PRINT
003234  012124                                   MES24                      ;SWITCH VOLTAGE NEG.
003236  104012                                   WAITGN
003240  000000                                   G1                         ;GAIN X1
003242  013212                                   NEG625                     ;SHOULD = -0.625V
003244  013274                                   GM62X1


                                        ;TEST -0.625V X G2
003246  104013                                   TAKEGN
003250  020000                                   G2                         ;GAIN X2
003252  013210                                   NEG125                     ;SHOULD = -1.25V
003254  013304                                   GM62X2                     ;SAVE IT

```

```
                                        ;TEST -0.625V X G4
003256  104013                                  TAKEGN
003260  040000                                  G4                              ;GAIN X4
003262  013206                                  NEG250                          ;SHOULD = -2.5V
003264  013314                                  GM62X4

                                        ;TEST -0.625V X G8
003266  104013                                  TAKEGN
003270  060000                                  G8                              ;GAIN X8
003272  013204                                  NEG500                          ;SHOULD = -5.00V
003274  013324                                  GM62X8

                                        ;TEST +0.3125V X G1
003276  104000                          GT5:    PRINT
003300  012167                                  MES28                           ;TEXT '+0.3125V'
003302  104012                                  WAITGN
003304  000000                                  G1                              ;GAIN X1
003306  013202                                  POS312                          ;SHOULD = +0.3125V
003310  013226                                  GP31X1                          ;SAVE IT

                                        ;TEST +0.3125V X G2
003312  104013                                  TAKEGN
003314  020000                                  G2                              ;GAIN X2
003316  013200                                  POS625                          ;SHOULD = +0.625V
003320  013236                                  GP31X2

                                        ;TEST +0.3125V X G4
003322  104013                                  TAKEGN
003324  040000                                  G4                              ;GAIN X4
003326  013176                                  POS125                          ;SHOULD = +1.25V
003330  013246                                  GP31X4

                                        ;TEST +0.3125V X G8
003332  104013                                  TAKEGN
003334  060000                                  G8                              ;GAIN X8
003336  013174                                  POS250                          ;SHOULD = +2.50V
003340  013256                                  GP31X8
003342  005767  007422                          TST     ADSIGN
003346  001422                                  BEQ     GT6                     ;BRANCH IS UNIPOLAR

                                        ;TEST -0.3125V X G1
003350  104000                                  PRINT
003352  012124                                  MES24                           ;TEXT 'SWITCH NEG.'
003354  104012                                  WAITGN
003356  000000                                  G1                              ;GAIN X1
003360  013214                                  NEG312                          ;SHOULD = -0.3125V
003362  013276                                  GM31X1

                                        ;TEST -0.3125V X G2
003364  104013                                  TAKEGN
003366  020000                                  G2                              ;GAIN X2
003370  013212                                  NEG625                          ;SHOULD = -0.625V
003372  013306                                  GM31X2
```

```
                        ;TEST -0.3125V X G4
003374  104013                  TAKEGN
003376  040000                  G4                      ;GAIN X4
003400  013210                  NEG125                  ;SHOULD = -1.25V
003402  013316                  GM31X4

                        ;TEST -0.3125V X G8
003404  104013                  TAKEGN
003406  060000                  G8                      ;GAIN X8
003410  013206                  NEG250                  ;SHOULD = -2.50V
003412  013326                  GM31X8

                        ;TEST +0.1563V X G2
003414  104000          GT6:    PRINT
003416  012200                  MES29                   ;TEXT '+0.1563V'
003420  104012                  WAITGN
003422  020000                  G2                      ;GAIN X2
003424  013202                  POS312                  ;SHOULD = +0.3125V
003426  013240                  GP15X2

                        ;TEST +0.1563V X G4
003430  104013                  TAKEGN
003432  040000                  G4                      ;GAIN X4
003434  013200                  POS625                  ;SHOULD = +0.625V
003436  013250                  GP15X4

                        ;TEST +0.1563V X G8
003440  104013                  TAKEGN
003442  060000                  G8                      ;GAIN X8
003444  013176                  POS125                  ;SHOULD = +1.25V
003446  013260                  GP15X8
003450  005767  007314          TST     ADSIGN
003454  001416                  BEQ     GT7             ;BRANCH IF UNIPOLAR

                        ;TEST -0.1563V X G2
003456  104000                  PRINT
003460  012124                  MES24                   ;TEXT 'SWITCH NEG.
003462  104012                  WAITGN
003464  020000                  G2                      ;GAIN 'X2'
003466  013214                  NEG312                  ;SHOULD = -0.3125V
003470  013310                  GM15X2

                        ;TEST -0.1563V X G4
003472  104013                  TAKEGN
003474  040000                  G4                      ;GAIN X4
003476  013212                  NEG625                  ;SHOULD = -0.625V
003500  013320                  GM15X4

                        ;TEST -0.1563V X G8
003502  104013                  TAKEGN
003504  060000                  G8                      ;GAIN X8
003506  013210                  NEG125                  ;SHOULD = -1.25V
003510  013330                  GM15X8

```

```
                                  ;TEST +0.0781V X G4
003512  104000                    GT7:    PRINT
003514  012211                            MES30                     ;TEXT '+0.0781V'
003516  104012                            WAITGN
003520  040000                            G4                        ;GAIN X4
003522  013202                            POS312                    ;SHOULD = +0.3125V
003524  013252                            GPO7X4


                                  ;TEST +0.0781V X 8
003526  104013                            TAKEGN
003530  060000                            G8                        ;GAIN X8
003532  013200                            POS625                    ;SHOULD = +0.625V
003534  013262                            GPO7X8
003536  005767  007226                    TST     ADSIGN
003542  001412                            BEQ     GT8


                                  ;TEST -0.0781V X G4
003544  104000                            PRINT
003546  012124                            MES24                     ;TEXT 'SWITCH NEG.'
003550  104012                            WAITGN
003552  040000                            G4                        ;GAIN X4
003554  013214                            NEG312                    ;SHOULD = -0.3125V
003556  013222                            GMO7X4


                                  ;TEST -0.0781V X G8
003560  104013                            TAKEGN
003562  060000                            G8                        ;GAIN X8
003564  013212                            NEG625                    ;SHOULD = -0.625V
003566  013332                            GMO7X8


                                  ;TEST +0.0390V X G8
003570  104000                    GT8:    PRINT
003572  012222                            MES31                     ;TEXT '+0.0390V'
003574  104012                            WAITGN
003576  060000                            G8                        ;GAIN X8
003600  013202                            POS312                    ;SHOULD = +0.3125V
003602  013264                            GPO3X8
003604  005767  007160                    TST     ADSIGN
003610  001406                            BEQ     GT9


                                  ;TEST -0.0390V X G8
003612  104000                            PRINT
003614  012124                            MES24                     ;TEXT 'SWITCH NEG.'
003616  104012                            WAITGN
003620  060000                            G8
003622  013214                            NEG312                    ;SHOULD = -0.3125V
003624  013334                            GMO3X8
```

```

                                                ;TYPE OUT A HISTOGRAM OF ALL THE GAIN AVERAGES.

003626  012767  013216  003302  GT9:     MOV     #GPSDX1,AVGTAB  ;SET UP GAIN TABLE
003634  012767  000002  007202           MOV     #2,KSTOR1
003642  104000                           PRINT
003644  012233                           MES32                   ;TYPE TABLE 'HEADER'.
003646  012767  000005  007150           MOV     #5,ICOUNT       ;SET UP PRINT ROUTINE
003654  104000                  GT10:    PRINT
003656  012370                           MES34                   ;TYPE GAIN X1 VALUES
003660  104014                           PRTAVG                  ;TYPE OUT AVERAGES X1
003662  104000                           PRINT
003664  012400                           MES35                   ;TYPE GAIN X2
003666  012767  000001  007126           MOV     #1,COUNT
003674  104015                           SIXDSH                  ;TYPE DASHES
003676  104014                           PRTAVG                  ;TYPE OUT AVERAGES X2
003700  104000                           PRINT
003702  012410                           MES36
003704  012767  000002  007110           MOV     #2,COUNT
003712  104015                           SIXDSH                  ;TYPE DASHES
003714  104014                           PRTAVG                  ;TYPE OUT AVERAGES X4
003716  104000                           PRINT
003720  012420                           MES37
003722  012767  000003  007072           MOV     #3,COUNT
003730  104015                           SIXDSH
003732  104014                           PRTAVG                  ;TYPE OUT AVERAGES X8
003734  005767  007030                   TST     ADSIGN
003740  001002                           BNE     GT11            ;IF UNIPOLAR, TYPE OUT NEG. COUNTS
003742  000167  176764                   JMP     GT0             ;OTHERWISE RESTART GAIN TEST
003746  005367  007072          GT11:    DEC     KSTOR1
003752  001002                           BNE     .+6
003754  000167  176752                   JMP     GT0
003760  104000                           PRINT
003762  011453                           CRLF
003764  000733                           BR      GT10
003766  104011                  XWATGN:  TTYIN                   ;WAIT FOR 'CR' BEFORE CONTINUING
003770  005067  007076                   CLR     RETSWH          ;CLR SOFTWARE SW.
003774  042767  060000  007032           BIC     #60000,INITAL   ;CLR GAIN BITS
004002  057667  000000  007024           BIS     @(SP),INITAL    ;GET SPECIFIED GAIN AND SET IT UP.
004010  017667  000000  007026           MOV     @(SP),KSTOR1    ;SAVE GAIN
004016  062716  000002                   ADD     #2,(SP)         ;SET UP THE ADDRESS OF TRUE VOLTAGE
004022  017667  000000  000226           MOV     @(SP),PRTADR    ;SAVE ADDRESS JF TRUE VOLTAGE
004030  062716  000002                   ADD     #2,(SP)         ;SET UP STORAGE ADDRESS FOR VOLTAGE
004034  017667  000000  007004           MOV     @(SP),KSTOR2    ;SAVE ADDRESS
004042  062716  000002                   ADD     #2,(SP)         ;SET UP STACK TO EXIT
004046  104016                  GLOOP:   TSTTKS                  ;TEST FOR KEYBOARD FLAG
004050  012777  176000  175146           MOV     #-2000,@ADWCR   ;SET UP TO TAKE '1024' CONVERSIONS
004056  004767  002012                   JSR     PC,ADCNVT       ;TAKE THE CONVERSIONS
004062  104004                           CMPUTE                  ;COMPUTE THE AVERAGE
004064  016777  007024  006754           MOV     AVRAGE,@KSTOR2  ;SAVE THE AVERAGE VALUE
004072  104005                           CATORIZ                 ;CATAGORIZE THE COUNT SPREAD
004074  026777  007014  000154           CMP     AVRAGE,@PRTADR  ;IS AVERAGE =TO KNOWN VALUE?
004102  001414                           BEQ     GANEXT          ;EXIT IF EQUAL
004104  026777  007006  000144           CMP     AVERP1,@PRTADR  ;IS AVERAGE =TO KNOWN VALUE +1?
004112  001410                           BEQ     GANEXT          ;EXIT IF EQUAL
004114  027767  000136  006770           CMP     @PRTADR,AVERM1  ;IS AVERAGE =TO KNOWN VALUE -1?
```

```
004122  001404                           BEQ     GANEXT          ;EXIT IF EQUAL
004124  032777  040000  175060  GAINER:  BIT     #SW14,@SWR      ;TEST FOR INHIBIT SCOPE LOOPING
004132  001406                           BEQ     GERR1           ;BRANCH IF NOT SET
004134  005767  006732          GANEXT:  TST     RETSWH          ;WAS AN ERROR REPORTED?
004140  001402                           BEQ     .+6             ;NO, CONTINUE
004142  104000                           PRINT
004144  011453                           CRLF
004146  000002                           RTI
004150  032777  020000  175034  GERR1:   BIT     #SW13,@SWR      ;TEST FOR PRINT INHIBIT
004156  001333                           BNE     GLOOP           ;BRANCH IS SW SET
004160  005767  006710                   TST     MESPRT          ;TEST IF TITLE HAS BEEN TYPED
004164  001005                           BNE     GERR2           ;BRANCH IF YES
004166  052767  000001  006700           BIS     #1,MESPRT       ;OTHERWISE TYPE ERROR HEADER
004174  104000                           PRINT
004176  012340                           MES33                   ;TEXT 'GAIN
004200  005767  006640          GERR2:   TST     KSTOR1          ;TEST FOR 'G1'
004204  001003                           BNE     GERR3           ;BRANCH IF NOT
004206  104000                           PRINT
004210  012370                           MES34                   ;TEXT '1'
004212  000420                           BR      GERR6
004214  022767  020000  006622  GERR3:   CMP     #G2,KSTOR1      ;TEST FOR GX2
004222  001003                           BNE     GERR4           ;BRANCH IF NOT -
004224  104000                           PRINT
004226  012400                           MES35                   ;TEXT '2'
004230  000411                           BR      GERR6
004232  022767  040000  006604  GERR4:   CMP     #G4,KSTOR1      ;TEST FOR GAIN OF '4'
004240  001003                           BNE     GERR5           ;BRANCH AND PRINT GX8
004242  104000                           PRINT
004244  012410                           MES36                   ;TEXT '4'
004246  000402                           BR      GERR6
004250  104000                  GERR5:   PRINT
004252  012420                           MES37                   ;TEXT '8'
004254  104010                  GERR6:   PRTOCT
004256  000000                  PRTADR:  0                       ;TYPE VOLTAGE VALUE
004260  104007                           SPACE                   ;TYPE SPACE
004262  104007                           SPACE
004264  104010                           PRTOCT
004266  013114                           AVRAGE                  ;TYPE AVERAGE
004270  005267  006576                   INC     RETSWH
004274  000664                           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.

 004276  012767  004310  006474  RECVRY: MOV     #RECVY1,AVECTR  ;SET UP THE '↑A' RETURN ADDRESS
 004304  104000                          PRINT
 004306  011540                          MES8                    ;TEXT 'RECOVERY TEST'
 004310  012767  000010  006506  RECVY1: MOV     #10,ICOUNT      ;SET UP TO PRINT '8' VALUES
 004316  005067  006512                  CLR     INITAL          ;CLR CH. TEMP. STORAGE
 004322  005067  006512                  CLR     INITL2          ;CLR 2ND CH. STORAGE
 004326  012767  004354  006446          MOV     #RECVY2,PVECTR
 004334  052767  010000  006472          BIS     #10000,INITAL   ;SELECT RANDOM,DMA
 004342  052767  010000  006470          BIS     #10000,INITL2
 004350  104017                          GETCHA                  ;REQUEST CHANNELS
 004352  104003                          GAININ                  ;GET THE GAIN SETTINGS
 004354  012767  000020  006440  RECVY2: MOV     #20,COUNT       ;TAKE '16' CONVERSIONS
 004362  012701  013336                  MOV     #RANBUF,R1      ;SET UP RANDOM TABLE
 004366  012702  000010                  MOV     #10,R2          ;SAVE 8 VALUES
 004372  016721  006436          RECVY3: MOV     INITAL,(R1)+
 004376  005302                          DEC     R2
 004400  001374                          BNE     RECVY3
 004402  012702  000015                  MOV     #15,R2
 004406  016721  006426          RECVY4: MOV     INITL2,(R1)+
 004412  005302                          DEC     R2
 004414  100374                          BPL     RECVY4
 004416  104016                  RECVY5: TSTTKS                  ;CHECK FOR KEYBOARD FLAG
 004420  012777  177760  174576          MOV     #-20,@ADWCR     ;SET W.C FOR '16' CONVERSIONS
 004426  004767  001442                  JSR     PC,ADCNVT       ;TAKE THE CONVERSIONS
 004432  032777  020000  174552          BIT     #SW13,@SWR      ;TEST THE PRINT INHIBIT SW
 004440  001366                          BNE     RECVY5          ;BRANCH IF SET
 004442  104000                          PRINT
 004444  011561                          MES9                    ;TEXT 'CH.'
 004446  016702  006370                  MOV     FINAL2,R2
 004452  104006                          BINDEC                  ;TYPE 2ND CH.
 004454  104007                          SPACE
 004456  012767  013454  002452          MOV     #ADBUFF+20,AVGTAB
 004464  104014                          PRTAVG                  ;PRINT VALUES OF 2ND CH.
 004466  000753                          BR RECVY5               ;DO IT AGAIN
```

```



                                                 ;*******************************************************
                                                 ;INCREMENT MEMORY TEST
                                                 ;*******************************************************

                                                 ;THIS TEST IS DESIGNED TO TAKE CONVERTIONS ONE AT A TIME
                                                 ;IN THE SINGLE CHANNEL MODE USING INCREMENT MEMORY. THE
                                                 ;SWAR REGISTER IS READ AFTER EACH CONVERTION AND THE SPECIFIED
                                                 ;ADDRESS IS TESTED FOR MODIFICATION. THEM THE CONTENTS OF THE
                                                 ;SWAR IS PRINTED OR DISPLAYED.IF SWITCH REG. BIT15=1 THE INCREMEMTED
                                                 ;ADDRESS IS DISPLAYED THROUGH RO OTHERWISE PRINTED ON THE TELETYPE
                                                 ;THE OPERATOR MAY CHECK THAT EACH ADDRESS LINE IS SELECTABLE

004470  012777  000340  174502  INCTST: MOV     #340,@PSW
004476  016706  006274                  MOV     STACK,SP
004502  005067  006304                  CLR     SOFLAG          ;CLEAR PRINT FLAG
004506  012767  004634  006266          MOV     #RETA,PVECTR            ; SET UP A CONTROL P START
004514  012767  004544  006256          MOV     #INCTSA,AVECTR          ;SET UP A RESTART ADDRESS
004522  005767  006256          INCTSB: TST     INCFLG                  ;TEST IF CORE AVAIL
004526  001004                          BNE     1$
004530  104000                          PRINT
004532  012502                          MES42                   ; REPORT INSUFFICIENT CORE
004534  000167  175026                  JMP     MONITR          ; RETURN TO MONITOR
004540  104000                  1$:     PRINT
004542  012566                          MES44                   ;TEXT INSUFFICIENT CORE
004544  104000                  INCTSA: PRINT
004546  011567                          MES10                   ;SYNC I OR E
004550  104011                          TTYIN                   ; 1 CHARACTER
004552  016767  001112  006236          MOV     INBUF,PROC      ;SAVE IN TEMP STORE
004560  104000                          PRINT
004562  011453                          CRLF
004564  104000  012614          1$:     PRINT,MES45                     ;ASK FOR TEST
004570  104011                          TTYIN                   ;RECIEVE 1 CHAR.
004572  016767  001072  006230          MOV     INBUF,ITEST     ;SAVE TTY CHAR
004600  022767  000123  006222          CMP     #123,ITEST              ;IS IT AN S?
004606  001407                          BEQ     2$
004610  022767  000115  006212          CMP     #115,ITEST              ;IS IT A M?
004616  001525                          BEQ     QUANT           ;GO TO QUANTITIVE TEST
004620  104000  011463                  PRINT,QMARK             ;TAINT EITHER
004624  000757                          BR      1$              ;TRY AGAIN
004626  012767  000011  006214  2$:     MOV     #11,KSTOR3              ;PRINT 9/LINE
004634  012777  020000  174356  RETA:   MOV     #20000,@ADCSR   ;CLR ALL FLAGS
004642  004767  000460                  JSR     PC,CLRCOR       ;CLR CLEAR CORE FROM OFFSET-MEMSIZ
004646  017767  174340  006156          MOV     @SWR,STA        ;GET CH. FROM S.R
004654  042767  177000  006150          BIC     #177000,STA     ;CLR UNWANTED BITS
004662  052767  113000  006142          BIS     #113000,STA     ;SEQ,DMA,FINAL CH.,INC.MEM.
004670  022767  000105  006120          CMP     #105,PROC       ;TEST SYNC SEL
004676  001003                          BNE     .+10    ; BRANCH IF NOT EXTERNAL
004700  052767  004000  006124          BIS     #4000,STA       ;EST EX SYNC
004706  016767  006120  006122          MOV     STA,FINAL       ;LOAD VALUE OF REG
004714  042767  001000  006110          BIC     #1000,STA       ;CLR FINAL CHANNEL
004722  016767  006104  006104          MOV     STA,INITAL      ;LOAD VALUE OF INITIAL REG
004730  016777  006054  174276          MOV     OFFSET,@ADADR   ;LOAD OFFSET REGISTER
004736  012777  177777  174260          MOV     #-1,@ADWCR      ;TAKE ONE CONVERTION
004744  004767  001124                  JSR     PC,ADCNVT       ;;DO IT
```

```
004750 017767 174246 006054             MOV     @ADSWR,STA          ;GET ADDRESS OF INCREMENTED LOC
004756 022777 000001 006046             CMP     #1,@STA             ;WAS LOCATION MODIFIED BY +1
004764 001411                           BEQ     GDVAL
004766 104000                           PRINT            ; 'CORE LOCATION NOT INCREMENTED'
004770 012527                           MES43
004772 017767 174224 006026             MOV     @ADSWR,HOLD         ;SETUP ADDRESS TO PRINT
005000 104010                           PRTOCT                      ;VALUE OF SWAR
005002 013026                           HOLD
005004 000167 177624                    JMP     RETA                ;TRY AGAIN
005010 104016                   GDVAL:  TSTTKS
005012 005777 174174                    TST     @SWR                ;TEST BIT 15 FOR PRINT
005016 100405                           BMI     .+14
005020 017700 174176                    MOV     @ADSWR,R0           ;PRINT
005024 000005                           RESET
005026 000167 177602                    JMP     RETA                ;NO PRINT RETURN TO RESTART
005032 017767 174164 005766             MOV     @ADSWR,HOLD         ;GET CONTENTS OF SWAR
005040 104010                           PRTOCT
005042 013026                           HGLD
005044 104007                           SPACE           ;PRINT ONE BY DEFAULT
005046 005367 005776                    DEC     KSTOR3              ;TEST FOR END OF LINE
005052 001005                           BNE     .+14
005054 012767 000011 005766             MOV     #11,KSTOR3          ;RESET
005062 104000                           PRINT
005064 011453                           CRLF
005066 000167 177542                    JMP     RETA
                                        ;
                                        ;TAKE INCREMENT MEMORY CONVERTIONS AND OUTPUT CONVERTION
                                        ;RESULTS  SHOWING THE LOCATION INCREMENTED AND # OF INCREMENTS
                                        ;

005072 004767 000230            QUANT:  JSR     PC,CLACOR           ;CLEAR CORE FROM OFFSET TO MEMSIZE
005076 012777 020000 174114             MOV     #20000,@ADCSR       ;CLR ALL FLAGS
005104 017767 174102 005720             MOV     @SWR,STA            ;GET CH. FROM SWITCH REG.
005112 042767 177000 005712             BIC     #177000,STA                 ;CLEAR UNWANTED BITS
005120 052767 113000 005704             BIS     #113000,STA         ;SEQ,DMA,FINAL CH,INC.MEM.
005126 022767 000105 005662             CMP     #105,PROC           ;TEST FOR SYNC SELECT
005134 001003                           BNE     .+10
005136 052767 004000 005666             BIS     #4000,STA           ;INSERT THE EXTERNAL
005144 016767 005662 005664             MOV     STA,FINAL           ;LOAD FINAL ADDRESS
005152 042767 001000 005652             BIC     #1000,STA           ;CLR FINAL BIT
005160 016767 005646 005646             MOV     STA,INITAL          ;LOAD VALUE OF INITIAL REG.
005166 016777 005616 174040             MOV     OFFSET,@ADADR       ;LOAD INC MEM OFFSET REG.
005174 012777 176030 174022             MOV     #-1750,@ADWCR       ;TAKE 1000 (DEC)CONVERTIONS
005202 004767 000666                    JSR     PC,ADCNVT           ;TAKE CONVERTIONS
                                ;
                                ;SCALE FROM OFFSET TO MEMSIZ REPORTING ON CONSOLE DEVICE
                                ;EACH TIME AN INCREMENTED LOCATION IS ENCOUNTERED WITH
                                ;THE FOLLOWING (XXXXXX   ,# OF CONERTIONS)WHERE XXXXXX=CORE MEMORY
                                ;LOCATION INCREMENTED AND # OF CONVERTIONS=# OF CONVERTIONS INCREMENTING
                                ;THE LOCATIONS
                                ;
005206 005767 005600                    TST     SOFLAG              ;HAS INITIAL MESSAGE BEEN TYPED
005212 100410                           BMI     1$
005214 104000 012746                    PRINT,MES47                 ;PRINT MEMORY OFFSET=
005220 104010                           PRTOCT                      ;PRINT OFFSET
005222 013010                           OFFSET
```

```
005224  104000  012661                      PRINT,MES46                 ;PRINT HEADER
005230  005367  005556                      DEC     SOFLAG              ;SET FLAG
005234  016701  005550              1$:     MOV     OFFSET,R1           ;POINT TO TABLE
005240  104016                              TSTTKS
005242  005711                      2$:     TST     (R1)                ;ANY INCS?
005244  001005                              BNE     4$
005246  020167  005534              5$:     CMP     R1,MEMSIZ           ;AT TOP OF TABLE YET?
005252  001420                              BEQ     3$                  ;IF YES GET OUT
005254  005721                              TST     (R1)+               ;UPDATE POINTER
005256  000771                              BR      2$                  ;KEEP CYCLEING
005260  010167  005542              4$:     MOV     R1,HOLD             ;LOAD VALUE FOR PRINTING
005264  104010                              PRTOCT                      ;PRINT THE LOCATION INCREMENTED
005266  013026                              HOLD
005270  012767  000003  000110              MOV     #3,SPACEX           ;PRINT 3 SPACES
005276  104007                              SPACE                       ;PRINT THEM
005300  104016                              TSTTKS                      ;TEST TTY STATUS
005302  011102                              MOV     (R1),R2             ;GET DATA AND RESTORE R1
005304  104006                              BINDEC                      ;CONVERT DATA IN R2 AND PRINT
005306  104000  011453                      PRINT,CRLF
005312  000755                              BR      5$                  ;CYCLE TILL DONE
005314  104000  011453              3$:     PRINT,CRLF                  ;DONE THIS LOOP
005320  104016                              TSTTKS                      ;TEST TTY STATUS
005322  000167  177544                      JMP     QUANT               ;KEEP RUNNING
                                    ;
                                    ;

```

```
                                        ;SUBROUTINE TO 'CLR' CORE BEFORE USINS THE 'INC. MEM. MODE'

005326  005767  005452                  CLRCOR: TST     INCFLG          ;MEMORY AVAILABLE?
005332  001410                                  BEQ     EXCORE          ;IF NOT EXIT
005334  016701  005450                          MOV     OFFSET,R1       ;START CLEARING CORE AT OFFSET
005340  005021                          CLRCR1: CLR     (R1)+           ;CLR CORE
005342  020167  005440                          CMP     R1,MEMSIZ       ;DONE
005346  001374                                  BNE     CLRCR1          ;BRANCH IF NOT
005350  005077  005432                          CLR     @MEMSIZ         ;CLR LAST LOCATION
005354  000207                          EXCORE: RTS     PC              ;EXIT

                                        ;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

005356  105777  173624                  XSPACE: TSTB    @TPS            ;WAIT FOR TTY READY
005362  100375                                  BPL     .-4
005364  012777  000240  173616                  MOV     #240,@TPB       ;OUTPUT A SPACE
005372  005367  000010                          DEC      SPACEX         ;DECREMENT COUNT
005376  003367                                  BGT     XSPACE          ;LOOP IF NOT DONE
005400  005067  000002                          CLR     SPACEX          ;LINECOUNT TO ZERO
005404  000002                                  RTI                     ;RETURN
005406  000000                          SPACEX: 0

                                        ;KEYBOARD SERVICE ROUTINE

005410  104020                          XTTYIN: SAVREG
005412  012704  005670                          MOV     #INBUF,R4       ;SETUP CHARACTER BUFFER
005416  005067  005376                          CLR     CHRCNT          ;CLEAR CHARACTER COUNTER
005422  005067  000244                          CLR     INBUF+2
005426  105777  173550                  INPUTA: TSTB    @TKS            ;CHARACTER READY?
005432  100375                                  BPL     INPUTA          ;NO, WAIT IT OUT
005434  017701  173544                          MOV     @TKB,R1         ;SAVE CHARACTER
005440  042701  000200                          BIC     #200,R1         ;STRIPE PARITY BIT
005444  120127  000060                          CMPB    R1,#60          ;IS IT A SPECIAL CHARACTER
005450  100420                                  BMI     SPCHR           ;YES, TEST IT
005452  122701  000137                          CMPB    #137,R1
005456  100415                                  BMI     SPCHR
005460  010124                          INPUTB: MOV     R1,(R4)+        ;SAVE CHARACTER
005462  005267  005332                          INC     CHRCNT          ;INCREMENT THE CHARACTER COUNT.
005466  022767  000007  005324                  CMP     #7,CHRCNT
005474  100472                                  BMI     SPCHRS          ;TYPE '?' IF TOO MANT CHAR.
005476  105777  173504                  OUTPTA: TSTB    @TPS            ;ECHO CHARACTER
005502  100375                                  BPL     OUTPTA
005504  110177  173500                          MOVB    R1,@TPB
005510  000746                                  BR      INPUTA          ;WAIT FOR NEXT CHARACTER
```

```
                                        ;TEST FOR SPECIAL CHARACTERS :'↑A','↑C','↑P','+','CR',',' OR 'RUBOUT'

005512  122701  000001          SPCHR:  CMPB    #1,R1           ;CHAR.='↑A'
005516  001006                          BNE     SPCHRP          ;NO, NOT '↑A'
005520  104000                          PRINT                   ;ECHO '↑A'
005522  011445                          CNTRLA                  ;RESTORE 'SP'
005524  022626                          POP2SP
005526  104021                          GETREG
005530  000177  005244                  JMP     @AVECTR         ;YES, EXIT VIA '↑A' VECTOR ADDRESS.
005534  022701  000020          SPCHRP: CMP     #20,R1          ;CHR. ='↑P'
005540  001006                          BNE     SPCHR1          ;NOT '↑P'
005542  022626                          POP2SP                  ;YES, RESTORE 'SP'
005544  104000                          PRINT
005546  011450                          CNTRLP                  ;TEST '↑P'
005550  104021                          GETREG
005552  000167  001222                  JMP     DCODER          ;EXIT TO COMMAND DECODER
005556  122701  000003          SPCHR1: CMPB    #3,R1           ;CHAR.='↑C'
005562  001002                          BNE     .+6             ;NO,NOT '↑C'
005564  000167  173776                  JMP     MONITR          ;YES, EXIT TO MONITOR
005570  122701  000177                  CMPB    #177,R1         ;CHAR. = 'RUBOUT'
005574  001011                          BNE     SPCHR2          ;IGNORE CHAR. & EXIT
005576  005767  005216                  TST     CHRCNT          ;IS RUBOUT LEGAL?
005602  001711                          BEQ     INPUTA          ;NO, IGNORE IT
005604  005367  005210                  DEC     CHRCNT
005610  012701  000134                  MOV     #134,R1         ;TYPE '\' TO INDICATE RUBOUUT
005614  005744                          TST     -(R4)           ;POP OFF LAST CHARACTER
005616  000727                          BR      OUTPTA          ;WAIT FOR NEXT CHARACTER
005620  122701  000053          SPCHR2: CMPB    #53,R1          ;TEST FOR '+'
005624  001004                          BNE     SPCHR3          ;BRANCH IF NO
005626  012767  000177  005134          MOV     #177,ADSIGN     ;YES, IDICATES UNIPOLAR
005634  000720                          BR      OUTPTA          ;WAIT NEXT CHAR.
005636  122701  000054          SPCHR3: CMPB    #54,R1          ;TEST FOR ','
005642  001706                          BEQ     INPUTB          ;LEGAL CHAR., SAVE IT
005644  122701  000015          SPCHR4: CMPB    #15,R1          ;=TO 'CARRIAGE RETURN' TO TERMINATE?
005650  001004                          BNE     SPCHR5          ;NO, CONTINUE
005652  104000                          PRINT                   ;YES, TYPE 'CR-LF'
005654  011453                          CRLF
005656  104021                  EXTTY:  GETREG
005660  000002                          RTI
005662  104000                  SPCHR5: PRINT                   ;OTHERWISE TYPE '?'
005664  011463                          QMARK
005666  000750                          BR XTTYIN               ;WAIT FOR NEW ENTRY
005670  000000                  INBUF:  0                       ;CHARACTER STORAGE BUFFER
        005710                          .=.+16
```

```
                                       ;SUBROUTINE WILL CONVERT 'N' BCD WORDS (SEPARATED VIA COMMA'S)
                                       ;WHICH WERE STORED IN A TABLE VIA 'TTYIN' TO OCTAL AND STORE THEM.

005710  104020                         BCDBIN: SAVREG
005712  104011                                 TTYIN                           ;INPUT & STORE DECIMAL VALUE
005714  012704  005670                         MOV     #INBUF,R4               ;SETUP ASCII STORAGE TABLE
005720  012703  006064                         MOV     #BCDTAB,R3              ;TABLE FOR STORAGE OF CONVERTED WORDS
005724  005067  000136                         CLR     BCDTAB+2
005730  005001                         BCDBN1: CLR     R1                      ;REG. TO STORE RUNNING TOTAL
005732  005002                                 CLR     R2                      ;TEMP. STORAGE FOR 'R1'
005734  005767  005060                 BCDBN2: TST     CHRCNT                  ;END OF DATA?
005740  003426                                 BLE     BCDEND                  ;YES, EXIT
005742  005367  005052                         DEC     CHRCNT                  ;DECREMENT CHARACTER COUNTER
005746  122714  000054                         CMPB    #54,(R4)                ;IS CHARACTER = TO ','?
005752  001421                                 BEQ     BCDEND                  ;YES, DECODE NEW WORD
005754  121427  000060                         CMPB    (R4),#60
005760  002435                                 BLT     BCDERR                  ;TEST FOR LEGAL NO.
005762  021427  000071                         CMP     (R4),#71
005766  003032                                 BGT     BCDERR
005770  042714  177760                         BIC     #177760,(R4)            ;STRIPE NO. TO BCD
005774  012400                                 MOV     (R4)+,R0                ;SAVE NO. IN R0.
005776  010102                                 MOV     R1,R2                   ;SAVE CURRENT TOTAL
006000  006301                                 ASL     R1                      ;NX2
006002  006301                                 ASL     R1                      ;NX4
006004  006301                                 ASL     R1                      ;NX8
006006  060201                                 ADD     R2,R1                   ;NX9
006010  060201                                 ADD     R2,R1                   ;NX10
006012  060001                                 ADD     R0,R1                   ;N+NEW NO.
006014  000747                                 BR      BCDBN2
006016  005724                         BCDEND: TST     (R4)+                   ;UPDATE BUFFER
006020  010123                                 MOV     R1,(R3)+                ;SAVE CONVERTED VALUE & SETUP TO SAVE NEXT
006022  005767  004772                         TST     CHRCNT                  ;FINISHED?
006026  001340                                 BNE     BCDBN1                  ;NO, CONVERT NEXT WORD
006030  026727  000030  000777                 CMP     BCDTAB,#777             ;TEST IF NO. <511
006036  100006                                 BPL     BCDERR                  ;REPORT ERROR IF NOT
006040  026727  000022  000777                 CMP     BCDTAB+2,#777           ;TEST IF 2ND. NO. <511
006046  100002                                 BPL     BCDERR                  ;BRANCH IF NOT
006050  104021                                 GETREG
006052  000002                                 RTI                             ;YES, EXIT
006054  104000                         BCDERR: PRINT
006056  011463                                 QMARK                           ;TYPE '?'.
006060  000167  177626                         JMP     BCDBIN+2
006064  000000                         BCDTAB: 0                               ;OCTAL STORAGE TABLE
006066  000000                                 0
006070  000000                                 0
006072  000000                                 0
```

```
                                            ;SUBROUTINE TO 'N' NUMBER OF A/D CONVERSIONS USING EITHER SEQUENTIAL
                                            ;OR RANDOM MODE.  ROUTINE IS ENTERED WITH 'N' IN COUNT AND THE
                                            ;CHANNEL AND GAIN TO BE CONVERTED IN 'INITAL' & 'FINAL' ADDRESSES.

006074  052777  020000  173116  ADCNVT: BIS     #20000,@ADCSR           ;CLR ALL.
006102  016777  004702  173124          MOV     OFFSET,@ADADR           ;LOAD OFFSET REG.
006110  012777  013336  173104          MOV     #RANBUF,@ADSWR          ;LOAD STATUS WORD REGISTER
006116  012777  013434  173104          MOV     #ADBUFF,@ADWRA          ;LOAD A/D WORD REGISTER 'A'
006124  017777  173074  173100          MOV     @ADWCR,@ADWRB           ;OFFSET BUFFER 'B' VIA OF NC. OF CONVERSIONS
006132  005477  173074                  NEG     @ADWRB                  ;MAKE NO. POS.
006136  006377  173070                  ASL     @ADWRB                  ;OFFSET X2
006142  062777  013434  173062          ADD     #ADBUFF,@ADWRB          ;FROM THE 'A' BUFFER
006150  004767  002540                  JSR     PC,LDINTR               ;LOAD THE A/D INTERRUPT VECTOR.
006154  006210                          CNVT1                           ;TO INTERRUPT HERE
006156  052767  001000  004652          BIS     #1000,FINAL             ;SET UP FINAL CH.
006164  005767  004644                  TST     INITAL                  ;RUNNING SEQ. MODE?
006170  100003                          BPL     .+10                    ;NO, DON'T LOAD FINAL CH.
006172  016777  004640  173016          MOV     FINAL,@ADCR             ;LOAD FINAL CH.
006200  016777  004630  173010          MOV     INITAL,@ADCR            ;LOAD INITIAL CH. & ST. CONVERTER.
006206  000001                          WAIT                            ;WAIT FOR INTERRUPT.

                                ;ENTERED HERE ON THE INTERRUPT

006210  022626                  CNVT1:  POP2SP                          ;RE-SET STACK
006212  004767  002532                  JSR     PC,CLRINT               ;CLR INTR. ADDR.
006216  005777  172776                  TST     @ADCSR                  ;TEST ERROR BIT
006222  100002                          BPL     CNVT2                   ;BRANCH IF NOT SET
006224  104007                          PRINT                           ;OTHERWISE TYPE ERROR
006226  011620                          MES12                           ;TEST 'ERROR BIT SET'
006230  000207                  CNVT2:  RTS     PC                      ;EXIT

                                ;POWER FAIL HANDLER

006232  010046                  PWRFAL: MOV     R0,-(SP)
006234  010146                          MOV     R1,-(SP)
006236  010246                          MOV     R2,-(SP)
006240  010346                          MOV     R3,-(SP)
006242  010446                          MOV     R4,-(SP)
006244  010546                          MOV     R5,-(SP)
006246  016746  171552                  MOV     24,-(SP)
006252  010667  004540                  MOV     SP,PROC
006256  012767  006266  171540          MOV     #PWRUP,24
006264  000000                          HALT
```

```
                                        ;POWER UP HANDLER

     006266  012777  000340  172704  PWRUP:  MOV     #340,@PSW
     006274  016706  004516                  MOV     PROC,SP
     006300  012667  171520                  MOV     (SP)+,24
     006304  012605                          MOV     (SP)+,R5
     006306  012604                          MOV     (SP)+,R4
     006310  012603                          MOV     (SP)+,R3
     006312  012602                          MOV     (SP)+,R2
     006314  012601                          MOV     (SP)+,R1
     006316  012600                          MOV     (SP)+,R0
     006320  005005                          CLR     R5
     006322  005205                          INC     R5
     006324  001376                          BNE     .-2
     006326  104000                          PRINT
     006330  012065                          MES21
     006332  000167  173230                  JMP     MONITR

                                        ;SUBROUTINE TO REQUEST A CH.(S) INPUT FROM THE TELETYPE

     006336  104000                  XCHAIN: PRINT
     006340  011666                          MES14                   ;TEXT 'CH.(S)'
     006342  104001                          DECOCT                  ;CONVERT TO OCTAL
     006344  005067  004466                  CLR     FINAL           ;CLR CH. STORAGE
     006350  042767  001777  004456          BIC     #1777,INITAL
     006356  005067  004460                  CLR     FINAL2          ;CLR 2ND CH. STORAGE
     006362  042767  001777  004450          BIC     #1777,INITL2
     006370  016767  177470  004440          MOV     BCDTAB,FINAL    ;LOAD AS FINAL CH.
     006376  056767  177462  004430          BIS     BCDTAB,INITAL   ;LOAD INITAL CH.
     006404  016767  177456  004430          MOV     BCDTAB+2,FINAL2 ;LOAD AS 2ND FINAL CH.
     006412  056767  177450  004420          BIS     BCDTAB+2,INITL2 ;LOAD 2ND INITIAL CH.
     006420  022767  002240  004352          CMP     #REPT1,AVECTR   ;ENTERED FROM REPEATIBILTY TEST?
     006426  001007                          BNE     EXCHAN          ;NO, EXIT
     006430  005767  177432                  TST     BCDTAB+2        ;WAS A SECOND CH. ENTERED?
     006434  001404                          BEQ     EXCHAN          ;NO, EXIT
     006436  026767  004400  004372          CMP     FINAL2,FINAL    ;YES, IS 2ND CH. > 1ST CH.
     006444  103734                          BLO     XCHAIN          ;NO, ILLEGAL ENTRY
     006446  000002                  EXCHAN: RTI
```

```
                                        ;SUBROUTINE TO INPUT A 'GAIN FROM THE TELETYPE

006450  104000                          XGAINA: PRINT
006452  011720                                  MES18                   ;TEXT 'GAIN?'
006454  104001                                  DECOCT                  ;CONVERT TO OCTAL
006456  012701  013034                          MOV     #INITAL,R1
006462  012702  006064                          MOV     #BCDTAB,R2
006466  004767  000020                          JSR     PC,XGAINB
006472  005722                                  TST     (R2)+           ;SET UP BCDTAB+2
006474  005712                                  TST     (R2)            ;WAS A SECOND GAIN ENTERED?
006476  001404                                  BEQ     EXGAIN          ;NO, EXIT
006500  012701  013040                          MOV     #INITL2,R1      ;SET UP SECOND GAIN
006504  004767  000002                          JSR     PC,XGAINB
006510  000002                          EXGAIN: RTI

006512  042711  060000                  XGAINB: BIC     #60000,(R1)     ;CLR GAIN BITS
006516  022712  000001                          CMP     #1,(R2)         ;TEST FOR '1'
006522  001001                                  BNE     XGAIN2          ;IF NOT '1' TEST FOR '2'
006524  000207                                  RTS     PC

006526  022712  000002                  XGAIN2: CMP     #2,(R2)
006532  001003                                  BNE     XGAIN4
006534  052711  020000                          BIS     #G2,(R1)
006540  000207                                  RTS     PC

006542  022712  000004                  XGAIN4: CMP     #4,(R2)
006546  001003                                  BNE     XGAIN8
006550  052711  040000                          BIS     #G4,(R1)
006554  000207                                  RTS     PC

006556  022712  000010                  XGAIN8: CMP     #10,(R2)
006562  001003                                  BNE     NOGAIN          ;ILLEGAL ENTRY, TRY AGAIN
006564  052711  060000                          BIS     #G8,(R1)
006570  000207                                  RTS     PC

006572  005726                          NOGAIN: POP1SP                  ;RESET STACK
006574  000725                                  BR      XGAINA          ;ACCEPT NEW GAIN

                                        ;********************************************************************
                                        ;SUBROUTINE ENTERED ON AN ILLEGAL TRAP.  THE ROUTINE REPORTS WHERE IT
                                        ;TRAPPED 'FROM' AND WHERE IT TRAP 'TO'.
                                        ;********************************************************************

006576  011667  004252                  ERTRAP: MOV     (SP),TEMP1      ;SAVE LOCATION WHERE IT TRAPPED 'TO'
006602  022626                                  POP2SP
006604  011667  004246                          MOV     (SP),TEMP2      ;SAVE WHERE IT TRAPPED FROM.
006610  104000                                  PRINT
006612  012451                                  MES40                   ;TEXT 'ILLEGAL TRAP TO'
006614  162767  000004  004232                  SUB     #4,TEMP1
006622  104010                                  PRTOCT
006624  013054                                  TEMP1                   ;TYPE 'PC' TRAPPED TO
006626  104000                                  PRINT
006630  012473                                  MES41                   ;TEXT 'FROM'
006632  162767  000002  004216                  SUB     #2,TEMP2
006640  104010                                  PRTOCT
006642  013056                                  TEMP2                   ;TYPE WHERE IT TRAPPED FROM
```

```
006644  000167  172716                   JMP     MONITR          ;RETURN TO MONITOR
                                 ;**********************************************************************
                                 ;SUBROUTINE TO SAVE 'R1-R5' ON STACK
                                 ;**********************************************************************

006650  012667  004206           XSAVRG: MOV     (SP)+,SAVEPC
006654  012667  004204                   MOV     (SP)+,SAVPSW
006660  012667  004202                   MOV     (SP)+,SAV2PC
006664  012667  004200                   MOV     (SP)+,SAV2SW
006670  010146                           MOV     R1,-(SP)
006672  010246                           MOV     R2,-(SP)
006674  010346                           MOV     R3,-(SP)
006676  010446                           MOV     R4,-(SP)
006700  010546                           MOV     R5,-(SP)
006702  016746  004162                   MOV     SAV2SW,-(SP)
006706  016746  004154                   MOV     SAV2PC,-(SP)
006712  016746  004146                   MOV     SAVPSW,-(SP)
006716  016746  004140                   MOV     SAVEPC,-(SP)
006722  000002                           RTI

                                 ;**********************************************************************
                                 ;SUBROUTINE TO RESTORE 'R1-R5' FROM THE STACK
                                 ;**********************************************************************

006724  012667  004132           XGETRG: MOV     (SP)+,SAVEPC
006730  012667  004130                   MOV     (SP)+,SAVPSW
006734  012667  004126                   MOV     (SP)+,SAV2PC
006740  012667  004124                   MOV     (SP)+,SAV2SW
006744  012605                           MOV     (SP)+,R5
006746  012604                           MOV     (SP)+,R4
006750  012603                           MOV     (SP)+,R3
006752  012602                           MOV     (SP)+,R2
006754  012601                           MOV     (SP)+,R1
006756  016746  004106                   MOV     SAV2SW,-(SP)
006762  016746  004100                   MOV     SAV2PC,-(SP)
006766  016746  004072                   MOV     SAVPSW,-(SP)
006772  016746  004064                   MOV     SAVEPC,-(SP)
006776  000002                           RTI
```

```
                                ;COMMAND DECODER, ENTERED VIA '↑P'
                                ;ROUTINE ALLOWS THE USER TO CHANGE A SINGLE PARAMETER (CHANNEL, GAIN
                                ;COUNT SPREAD OR SYNC TYPE) WITHOUT CHANGING ALL PARAMETERS.

007000  104000                  DCODER: PRINT
007002  011455                          ASTRIC                  ;TYPE '*' TO INDICATE READY
007004  104011                          TTYIN                   ;WAIT FOR INPUT
007006  012701  005670                  MOV     #INBUF,R1       ;SET UP TO TEST CHAR.
007012  022711  000103                  CMP     #103,(R1)       ;WAS 'C' TYPED?
007016  001411                          BEQ     PARMC           ;BRANCH IF YES AND DECODE 'C' OR 'CS'
007020  022711  000107                  CMP     #107,(R1)       ;WAS 'G' TYPED?
007024  001413                          BEQ     PARMG           ;BRANCH IF YES AND DECODE 'GAIN'
007026  022711  000123                  CMP     #123,(R1)       ;WAS 'S' TYPED
007032  001412                          BEQ     PARMS           ;BRANCH IF YES AND DECODE 'SYNC' OR 'ST'
007034  104000                          PRINT                   ;ILLEGAL CALL
007036  011463                          QMARK                   ;TYPE '?'
007040  000757                          BR      DCODER          ;RESTART
007042  005767  176624          PARMC:  TST     INBUF+2         ;TEST FOR 2 CHARACTER INPUT
007046  001020                          BNE     PARMCS          ;BRANCH IF YES AND DECODE COUNT SPREAD
007050  104017                          GETCHA                  ;OTHERWISE REQUEST CH.
007052  000752                          BR      DCODER          ;WAIT NEXT INSTRUCTION
007054  104003                  PARMG:  GAININ                  ;REQUEST GAIN
007056  000750                          BR      DCODER          ;WAIT NEXT INSTRUCTION
007060  005767  176606          PARMS:  TST     INBUF+2         ;TEST FOR 2 CHARACTER INPUT
007064  001402                          BEQ     .+6             ;BRANCH IF NO
007066  000177  003710                  JMP     @PVECTR         ;OTHERWISE ASSUME 'ST' & EXIT
007072  104000                          PRINT
007074  012441                          MES39                   ;TEXT 'SYNC?'
007076  104011                          TTYIN                   ;WAIT INPUT
007100  016767  176564  003710          MOV     INBUF,PROC      ;SAVE INPUT
007106  000734                          BR      DCODER
007110  104000                  PARMCS: PRINT
007112  011701                          MES16                   ;TEXT 'COUNT SPREAD'
007114  104001                          DECOCT                  ;CONVERT TO OCTAL
007116  016767  176742  003724          MOV     BCDTAB,KSTOR3   ;SAVE IT
007124  000725                          BR      DCODER

                                ;SUBROUTINE TO TYPE OUT '5' AVERAGES FOR THE GAIN TEST HISTOGRAM.

007126  016767  003672  003724  XPRTAV: MOV     ICOUNT,TEMP3
007134  104010                  XPTA1:  PRTOCT                  ;PRINT OCTAL VALUE OF GAIN AVERAGE
007136  013216                  AVGTAB: GP50X1
007140  062767  000002  177770          ADD     #2,AVGTAB       ;UPDATE GAIN TABLE
007146  012767  000002  176232          MOV     #2,SPACEX
007154  104007                          SPACE                   ;TYPE '2' SPACES
007156  005367  003676                  DEC     TEMP3
007162  001364                          BNE     XPTA1           ;IF NOT DONE, PRINT NEXT AVG.
007164  000002                          RTI
```

```


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

007166  011646                  EMTSRV: MOV     (SP),-(SP)              ;GET PC FOR TO RETURN
007170  162716  000002                  SUB     #2,(SP)                 ;PC OF EMT
007174  017616  000000                  MOV     @(SP),(SP)              ;GET EMT
007200  005716                          TST     (SP)                    ;IS EMT VALID?
007202  001001                          BNE     EMTOK
007204  000000                          HALT                            ;INVALID EMT
007206  006316                  EMTOK:  ASL     (SP)                    ;MULTIPLY EMT ARG BY '2'
007210  042716  177001                  BIC     #177001,(SP)            ;CLEAR UNWANTED BITS
007214  062716  007226                  ADD     #EMTTAB,(SP)            ;POINTER TO SUBROUTINE ADDRESS
007220  017616  000000                  MOV     @(SP),(SP)              ;SUBROUTINE ADDRESS
007224  000136                          JMP     @(SP)+                  ;GO TO SUBROUTINE

                                ;EMT DISPATCH TABLE

007226  007306                  EMTTAB: TYPMES                          ;MESSAGE PRINT ROUTINE
007230  005710                          BCDBIN                          ;DECIMAL TO BINARY CONVERSION ROUTINE
007232  010246                          XRDMEM                          ;SUBROUTINE TO READ & CATEGORIZE INC. MEM. VALUES
007234  006450                          XGAINA                          ;REQUEST A 'GAIN' FROM THE TTY.
007236  007604                          CMPTE                           ;SUBROUTINE TO COMPUTE THE AVG
007240  007726                          CATORZ                          ;SUBROUTINE TO COMPUTE 'COUNT SPREAD'
007242  007404                          DECPRT                          ;SUBROUTINE TO CONVERT OCT TO DEC + PRINT
007244  005356                          XSPACE                          ;SUBROUTINE TO TYPE SPACES
007246  010576                          OCTPRT                          ;OCTAL PRINT ROUTINE
007250  005410                          XTTYIN                          ;TELEPRINTER SERVICE ROUTINE
007252  003766                          XWATGN                          ;GAIN TEST CONVERSION ROUTINE
007254  003770                          XWATGN+2                        ;GAIN TEST CONVERSION ROUTINE
007256  007126                          XPRTAV                          ;SUBROUTINE TO PRINT OUT THE GAIN AVERAGES
007260  007272                          DASH6                           ;SUBROUTINE TO TYPE OUT '6' DASHES
007262  010702                          TKSFLG                          ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
007264  006336                          XCHAIN                          ;SUBROUTINE TO DECODE A CHANNEL FROM TTY
007266  006650                          XSAVRG                          ;SUBROUTINE TO SAVE REG'S ON THE STACK
007270  006724                          XGETRG                          ;SUBROUTINE TO GET REG'S FROM THE STACK

                                ;SUBROUTINE TO TYPE OUT 'N' SETS OF SIX DASHES

007272  104000                  DASH6:  PRINT
007274  012430                          MES38
007276  005367  003520                  DEC     COUNT
007302  001373                          BNE     DASH6
007304  000002                          RTI

```

```


                                   ;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.

007306  104020                     TYPMES: SAVREG
007310  017602  000000                     MOV     @(SP),R2           ;GET THE MESSAGE ADDRESS FROM START
007314  062716  000002                     ADD     #2,(SP)            ;SET UP STACK TO EXIT
007320  105777  171662             TYPERA: TSTB    @TPS
007324  100375                             BPL     TYPERA             ;WAIT FOR TTY DONE
007326  122712  000100                     CMPB    #100,(R2)          ;TEST FOR '@'
007332  001002                             BNE     TYPER1             ;BRANCH IF NO EQUAL
007334  104021                             GETREG
007336  000002                             RTI                        ;OTHERWISE EXIT
007340  122712  000045             TYPER1: CMPB    #45,(R2)           ;TEST FOR '%'
007344  001403                             BEQ     TYPECL             ;IF = TYPE 'CR-LF'
007346  112277  171636             TYPER2: MOVB    (R2)+,@TPB         ;OUTPUT CHAR.
007352  000762                             BR      TYPERA
007354  012777  000015  171626     TYPECL: MOV     #15,@TPB           ;TYPE 'CR'
007362  105777  171620                     TSTB    @TPS
007366  100375                             BPL     .-4
007370  012777  000012  171612             MOV     #12,@TPB
007376  105722                             TSTB    (R2)+              ;INCREMENT BUFFER
007400  104016                             TSTTKS
007402  000746                             BR      TYPERA
                                   ;PRINT DECIMAL VALUE IN R2

007404  104020                     DECPRT: SAVREG
007406  012767  177774  000152             MOV     #-4,DIGCNT
007414  012767  007574  000150             MOV     #DECPNT+2,DECPNT
007422  012767  000240  000140             MOV     #240,ZERO
007430  012767  177777  000126     TYPT1:  MOV     #-1,DIGIT
007436  005267  000122             TYPT2:  INC     DIGIT
007442  167702  000124                     SUB     @DECPNT,R2
007446  100373                             BPL     TYPT2
007450  067702  000116                     ADD     @DECPNT,R2
007454  104016                             TSTTKS
007456  004767  000022                     JSR     PC,DECOUT
007462  005267  000100                     INC     DIGCNT
007466  001002                             BNE     TYPT3
007470  104021                             GETREG
007472  000002                             RTI
007474  062767  000002  000070     TYPT3:  ADD     #2,DECPNT
007502  000752                             BR      TYPT1
007504  005767  000054             DECOUT: TST     DIGIT
007510  001010                             BNE     DEC1
007512  022767  177777  000046             CMP     #-1,DIGCNT
007520  001404                             BEQ     DEC1
007522  016767  000042  000034             MOV     ZERO,DIGIT
007530  000406                             BR      DEC2
007532  012767  000260  000030     DEC1:   MOV     #260,ZERO
007540  052767  000260  000016             BIS     #260,DIGIT
007546  105777  171434             DEC2:   TSTB    @TPS
007552  100375                             BPL     .-4
007554  016777  000004  171426             MOV     DIGIT,@TPB
007562  000207                             RTS     PC
```

```
007564  000000                  DIGIT:  0
007566  000000                  DIGCNT: 0
007570  000240                  ZERO:   240
007572  007574                  DECPNT: .+2
007574  001750                          1000.
007576  000144                          100.
007600  000012                          10.
007602  000001                          1.

                                ;COMPUTE THE RESULTS OF '1024' CONVERSIONS AS HIGH, LOW AND AVERAGE

007604  012701  001777          CMPTE:  MOV     #1777,R1                ;SET UP TO COMPARE '1023' NUMBERS
007610  005000                          CLR     R0              ;CLR HI ORDER DIVIDEND
007612  012704  013434                  MOV     #ADBUFF,R4      ;SET UP DATA BUFFER ADDRESS
007616  012403                          MOV     (R4)+,R3        ;STORE 1ST VALUE AS AVERAGE
007620  010367  003252                  MOV     R3,HIGH         ;HIGH
007624  010367  003250                  MOV     R3,LOW          ;& LOW
007630  066703  003160                  ADD     ADSIZE,R3       ;ADD OFFSET TO AVERAGE
007634  012402                  GETDAT: MOV     (R4)+,R2
007636  020267  003234                  CMP     R2,HIGH         ;IS NEW NO. GREATER THAN OLD NO.
007642  003402                          BLE     TSLO            ;BRANCH IF NOT GREATER
007644  010267  003226                  MOV     R2,HIGH         ;OTHERWISE SAVE AS NEW HIGH
007650  020267  003224          TSLO:   CMP     R2,LOW
007654  003002                          BGT     TAGA
007656  010267  003216                  MOV     R2,LOW          ;OTHERWISE SAVE AS NEW LOW
007662  066702  003126          TAGA:   ADD     ADSIZE,R2       ;ADD OFFSET TO MAKE ALL NO. POS.
007666  060203                          ADD     R2,R3           ;ADD LOW ORDER
007670  005500                          ADC     R0              ;ADD CARRY TO HI ORDER
007672  005301                          DEC     R1
007674  001357                          BNE     GETDAT          ;1024 ADDITIONS?
007676  012701  000012                  MOV     #12,R1          ;YES, DIVIDE/1024
007702  006200                  AVGDAT: ASR     R0
007704  006003                          ROR     R3              ;SHIFT CARRY BIT INTO LO ORDER
007706  005301                          DEC     R1
007710  001374                          BNE     AVGDAT          ;DONE?
007712  005503                          ADC     R3              ;YES, ADD REMAINDER TO LO ORDER
007714  166703  003074                  SUB     ADSIZE,R3       ;SUBTRACT OFFSET TO OBTAIN REAL AVERAGE
007720  010367  003170                  MOV     R3,AVRAGE       ;SAVE AS AVERAGE
007724  000002                          RTI
```

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

007726  012701  000005          CATORZ: MOV     #5,R1
007732  016702  003156                  MOV     AVRAGE,R2               ;MOV AVER. TO WORK AREA
007736  066702  003052                  ADD     ADSIZE,R2               ;MAKE AVG. POS.
007742  012703  013116                  MOV     #AVERP1,R3              ;SETUP DISTRIBUTION TABLE (POS.)
007746  005202                  FILE1:  INC     R2                      ;A=A+1
007750  010213                          MOV     R2,(R3)                 ;SAVE A+1
007752  166723  003036                  SUB     ADSIZE,(R3)+            ;RESTORE ORIGINAL VALUE
007756  005301                          DEC     R1                      ;SAVED '5' COUNTS?
007760  001372                          BNE     FILE1                   ;BRANCH IF NO
                                ;SET UP TABLE OF AVG. -1 TO -5
007762  012701  000005                  MOV     #5,R1
007766  016702  003122                  MOV     AVRAGE,R2               ;MOV AVG. TO WORK AREA.
007772  066702  003016                  ADD     ADSIZE,R2
007776  012703  013114                  MOV     #AVRAGE,R3              ;SET UP DISTRIBUTION TABLE NEG.
010002  005302                  FILE2:  DEC     R2                      ;A=1-1
010004  010243                          MOV     R2,-(R3)                ;SAVE 'A-1'
010006  166713  003002                  SUB     ADSIZE,(R3)             ;RESTORE ORIGINAL NO. -1
010012  005301                          DEC     R1                      ;SAVED '5' COUNTS?
010014  001372                          BNE     FILE2                   ;BRANCH IF NO

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

010016  012703  013130                  MOV     #ORLOW,R3               ;CLEAR COUNTS
010022  005023                  CATR1:  CLR     (R3)+
010024  022703  013162                  CMP     #ORHIGH+2,R3            ;FINISHED?
010030  001374                          BNE     CATR1                   ;NO, CLEAR NEXT COUNTER
010032  012703  002001                  MOV     #2001,R3                ;COMPARE '1024' COUNTS
010036  012700  013434                  MOV     #ADBUFF,R0              ;SET UP A/D BUFFER
010042  005303                  CATR2:  DEC     R3
010044  001433                          BEQ     CATR5                   ;EXIT IF '0'
010046  012004                          MOV     (R0)+,R4
010050  026704  003052                  CMP     AVERP5,R4
010054  100421                          BMI     OVRHI
010056  020467  003020                  CMP     R4,AVERM5
010062  100421                          BMI     OVRLO
010064  005001                          CLR     R1
010066  012702  013102                  MOV     #AVERM5,R2
010072  022204                  CATR3:  CMP     (R2)+,R4
010074  001405                          BEQ     CATR4
010076  005201                          INC     R1
010100  022701  000013                  CMP     #13,R1
010104  001372                          BNE     CATR3
010106  000000                          HALT                            ;FATAL ERROR MR. BOUSE!
010110  006301                  CATR4:  ASL     R1                      ;MULTIPLY 'OFFSET' X2
010112  005261  013132                  INC     MINUS5(R1)
010116  000751                          BR      CATR2
010120  005267  003034          OVRHI:  INC     ORHIGH
010124  000746                          BR      CATR2
010126  005267  002776          OVRLO:  INC     ORLOW
010132  000743                          BR      CATR2
```

```

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

 010134  016767  003004  003020  CATRS:  MOV     AVGCNT,XSPRD1
 010142  066767  003000  003012          ADD     PLUS1,XSPRD1
 010150  066767  002766  003004          ADD     MINUS1,XSPRD1    ;=TO NO. COUNTS AT SPREAD OF '1'
 010156  016767  003000  003000          MOV     XSPRD1,XSPRD2
 010164  066767  002760  002772          ADD     PLUS2,XSPRD2
 010172  066767  002742  002764          ADD     MINUS2,XSPRD2    ;=TO NO. COUNTS AT SPREAD OF '2'
 010200  016767  002760  002760          MOV     XSPRD2,XSPRD3
 010206  066767  002740  002752          ADD     PLUS3,XSPRD3
 010214  066767  002716  002744          ADD     MINUS3,XSPRD3    ;=TO NO. COUNTS AT SPREAD OF '3'
 010222  016767  002740  002740          MOV     XSPRD3,XSPRD4
 010230  066767  002720  002732          ADD     PLUS4,XSPRD4
 010236  066767  002672  002724          ADD     MINUS4,XSPRD4    ;=TO NO. COUNTS AT SPREAD OF '4'
 010244  000002                          RTI                      ;EXIT

                                        ;SUBROUTINE TO COMPUTE THE HIGH, AVERAGE AND LOW VALUES THAT WERE STORED
                                        ;IN MEMORY VIA THE INCREMENT MEMORY MODE

 010246  016700  002536          XRDMEM: MOV     OFFSET,R0        ;START AT  4K OR 8K FOR 13 BITS+++++++++
 010252  005001                          CLR     R1               ;START=TO -10V
 010254  005002                          CLR     R2               ;SET UP AS HIORDER DIVIDEND
 010256  005003                          CLR     R3               ;SET UP AS LO ORDER DIVIDEND
 010260  005067  002612                  CLR     HIGH
 010264  005067  002610                  CLR     LOW
 010270  005067  002516                  CLR     SOFLAG           ;INITIAL HEADER INDICATOR
 010274  011004                  RDMEM1: MOV     (R0),R4 ;MOVE THE COUNTS TO TEMP (R4)
 010276  005304                  RDMEM2: DEC     R4               ;DECREMENT TEMP
 010300  100414                          BMI     NXADDR           ;BRANCH IF EMPTY
 010302  010167  002570                  MOV     R1,HIGH          ;SAVE AS HIGH
 010306  060103                          ADD     R1,R3            ;ADD VALUES WEIGHT TO LOW ORDER
 010310  005502                          ADC     R2               ;ADD CARRY TO HIGH ORDER
 010312  005767  002474                  TST     SOFLAG           ;CHECK IF LOW VALUE HAS BEEN SAVED
 010316  001367                          BNE     RDMEM2           ;OTHERWISE LOOP AGAIN
 010320  010167  002554                  MOV     R1,LOW           ;SAVE 1ST ENTRY AS LOW VALUE
 010324  005367  002462                  DEC     SOFLAG           ;SET  FLAG TO INDICATE THAT LOW VALUE IS FILLED
 010330  000762                          BR      RDMEM2

 010332  005201                  NXADDR: INC     R1               ;INCREMENT VALUE
 010334  020067  002446                  CMP     R0,MEMSIZ        ;DONE?
 010340  001402                          BEQ     1$
 010342  005720                          TST     (R0)+            ;UPDATE POINTER
 010344  000753                          BR      RDMEM1
 010346  012704  000012          1$:     MOV     #12,R4           ;SET UP TP DIVIDE BY 1024
 010352  006202                  AVMEM:  ASR     R2
 010354  006003                          ROR     R3               ;SHIFT CARRY INTO LOW ORDER
 010356  005304                          DEC     R4               ;DONE?
 010360  001374                          BNE     AVMEM            ;BRANCH IF NO.
 010362  005503                          ADC     R3               ;YES, ADD REMAINDER TO LOW ORDER
 010364  010367  002524                  MOV     R3,AVRAGE        ;SUBTRACT OFFSET TO=REAL VALUE
 010370  005767  002374                  TST     ADSIGN           ;IF BI POLAR START FUDGEING THE SIGN
 010374  001437                          BEQ     6$               ;BITS TO COMPENSATE FOR LINEAR MEMORY
 010376  016704  002412                  MOV     ADSIZE,R4        ;ADSIZE TO DETERMINE IF SIGN IS SET
 010402  000241                          CLC
 010404  006304                          ASL     R4               ;1 LEFT FOR BEGINING FF SIGN BIT
```

```
010406  030467  002464                    BIT     R4,HIGH         ;IS SIGN BIT SET IN HIGH VALUE?
010412  001004                            BNE     1$
010414  066767  002352  002454            ADD     SIGEXT,HIGH     ;NO- THEREFORE ADD SIGN EXT
010422  000402                            BR      2$
010424  040467  002446          1$:       BIC     R4,HIGH ;YES- THEREFORE CLEAR IT
010430  030467  002444          2$:       BIT     R4,LOW  ;IS SIGN BIT SET IN LOW VALUE?
010434  001004                            BNE     3$
010436  066767  002330  002434            ADD     SIGEXT,LOW      ;NO-THEREFORE ADD SIGN EXT
010444  000402                            BR      4$
010446  040467  002426          3$:       BIC     R4,LOW  ;YES-THEREFORE CLEAR IT
010452  030467  002436          4$:       BIT     R4,AVRAGE       ;IS SIGN BIT SET IN AVE
010456  001004                            BNE     5$              ;IF SET CLEAR IT
010460  066767  002306  002426            ADD     SIGEXT,AVRAGE   ;NO-THEREFORE ADD SIGN BIT
010466  000402                            BR      6$
010470  040467  002420          5$:       BIC     R4,AVRAGE       ;YES-THEREFORE CLEAR IT
010474  005067  002430          6$:       CLR     ORLOW
010500  005067  002454                    CLR     ORHIGH
010504  006303                            ASL     R3              ;COMPENSATE FOR ADDRESSING IN PDP11
010506  066703  002276                    ADD     OFFSET,R3       ;ADD OFFSET TO GET ADDRESS
010512  162703  000012                    SUB     #12,R3          ;=TO AVG -5
010516  012701  013132                    MOV     #MINUS,R1       ;SET UP TO SAVE COUNTS
010522  020367  002262          AVMEM1:   CMP     R3,OFFSET
010526  002003                            BGE     SET
010530  005021                            CLR     (R1)+           ;FILL WITH 0
010532  005723                            TST     (R3)+           ;UPDATE POINTER
010534  000772                            BR      AVMEM1
010536  011321                  SET:      MOV     (R3),(R1)+      ;RETRIEVE DATA
010540  020367  002242                    CMP     R3,MEMSIZ       ;AT END OF TABLE YET?
010544  001407                            BEQ     2$
010546  020127  013160                    CMP     R1,#ORHIGH      ;
010552  001402                            BEQ     1$
010554  005723                            TST     (R3)+           ;UPDATE POINTER
010556  000767                            BR      SET
010560  000167  177350          1$:       JMP     CATRS
010564  020127  013160          2$:       CMP     R1,#ORHIGH      ;IF AT BELOW CORE TABLE FILL BUCKETS /0'S
010570  001773                            BEQ     1$
010572  005021                            CLR     (R1)+
010574  000773                            BR      2$

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

010576  104020                  OCTPRT:   SAVREG
010600  017600  000000                    MOV     @(SP),R0        ;THE ADDRESS OF WORD TO BE TYPED
010604  062716  000002                    ADD     #2,(SP)         ;SET UP STACK TO EXIT
010610  012701  000006                    MOV     #6,R1
010614  012767  000376  000056            MOV     #376,MASK       ;MASK FOR FIRST BIT
010622  000401                            BR      .+4
010624  006110                  SHIFT:    ROL     (R0)
010626  006110                            ROL     (R0)
010630  006110                            ROL     (R0)
010632  111002                            MOVB    (R0),R2
010634  146702  000040                    BICB    MASK,R2
010640  052702  000260                    BIS     #260,R2
010644  104016                            TSTTKS
010646  132777  000200  170332            BITB    #200,@TPS
```

```
010654  100374                           BPL     .-6             ;WAIT FOR PRINTER READY
010656  110277  170326                   MOVB    R2,@TPB         ;PRINT CHAR.
010662  012767  000370  000010           MOV     #370,MASK       ;MASK FOR NEXT '5' DIGITS
010670  005301                           DEC     R1
010672  001354                           BNE     SHIFT
010674  104021                           GETREG
010676  000002                           RTI
010700  000376                   MASK:   376



                                 ;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET

010702  105777  170274           TKSFLG: TSTB    @TKS            ;FLAG SET?
010706  100001                           BPL     .+4             ;NO, EXIT
010710  104011                           TTYIN                   ;YES,INQUIRE
010712  000002                           RTI
                                 ;SUBROUTINE TO SET UP THE A/I VECTOR ADDR TO ENABLE INTERUPTS.

010714  017677  000000  170314   LDINTR: MOV     @(SP),@ADINT    ;LOAD INTERRUPT SERVICE ADDRESS
010722  062716  000002                   ADD     #2,(SP)         ;SET UP STACK TO EXIT
010726  012777  000340  170304           MOV     #340,@ADLVL     ;SET A/D INTR LEVEL @7
010734  012777  000100  170256           MOV     #100,@ADCSR     ;SET INTERRUPT ENABLE
010742  005077  170232                   CLR     @PSW            ;SET PROC. PRIORITY @0
010746  000207                           RTS     PC


                                 ;SUBROUTINE TO RESET THE A/D VECTOR ADDR TO HALT ON INTERRUPTS

010750  012777  000340  170222   CLRINT: MOV     #340,@PSW       ;RE-SET PROC. PRIORITY @7
010756  042777  000100  170234           BIC     #100,@ADCSR     ;CLR A/D INTR ENABLE
010764  016777  170250  170244           MOV     ADLVL,@ADINT
010772  005077  170242                   CLR     @ADLVL
010776  000207                           RTS     PC
```

```
                                         ;MESSAGES
011000     000                                    .BYTE
011001     045  042101  030506           TITLE:   .ASCII  '%ADF11 PART II, ANALOG DIAGNOSTIC TEST, 23/8/75'
011006  020061  040520  052122
011014  044440  026111  040440
011022  040516  047514  020107
011030  044504  043501  047516
011036  052123  041511  052040
011044  051505  026124  031040
011052  027463  027470  032467
011060  046450  044501  042116                    .ASCII  '(MAINDEC-11-DZADH-A)@'
011066  041505  030455  026461
011074  055104  042101  026510
011102  024501     100

011105     045  027501  020104           MES2:    .ASCII  '%A/D LENGTH? @'
011112  042514  043516  044124
011120  020077     100

011123     045  043442  044501           MES3:    .ASCII  '%"GAIN ACCURACY TEST". SUPPLY THE FOLLOWING VOLTAGES %'
011130  020116  041501  052503
011136  040522  054503  052040
011144  051505  021124  020056
011152  052523  050120  054514
011160  052040  042510  043040
011166  046117  047514  044527
011174  043516  053040  046117
011202  040524  042507  020123
011210     045
011211     040  047524  051440                    .ASCII  " TO SELECTED CH., TYPE 'CR' TO START TEST.@"
011216  046105  041505  042524
011224  020104  044103  026056
011232  052040  050131  020105
011240  041447  023522  052040
011246  020117  052123  051101
011254  020124  042524  052123
011262  040056

011264  052045  050131  020105           MES4:    .ASCII  "%TYPE LETTER ' ' TO RUN DESIRED TEST:%"
011272  042514  052124  051105
011300  023440  023440  052040
011306  020117  052522  020116
011314  042504  044523  042522
011322  020104  042524  052123
011330  022472
011332  041447  040447  044514                    .ASCII  "'C'ALIBRATION, 'R'EPEATIBILITY, 'G'AIN, R'E'COVERY,"
011340  051102  052101  047511
011346  026116  023440  023522
011354  050105  040505  044524
011362  044502  044514  054524
011370  020054  043447  040447
011376  047111  020054  023522
011404  023505  047503  042526
011412  054522     054
011415     047  023511  041516                    .ASCII  "'I'NCREMENT MEMORY%@"
011422  042522  042515  052115
```

```
011430  046440  046505  051117
011436  022531     100

011441     136  022503     100  CNTRLC: .ASCII  '↑C%@'

011445     136  040101          CNTRLA: .ASCII  '↑A@'

011450  050136     100          CNTRLP: .ASCII  '↑P@'

011453     045     100          CRLF:   .ASCII  '%@'

011455     045  040052          ASTRIC: .ASCII  '%*@'

011460  027045     100          DOT:    .ASCII  '%.@'

011463     077  040040          QMARK:  .ASCII  '? @'


011466  047111  027103  046440  MES6:   .ASCII  'INC. MEM. (Y OR N)? @'
011474  046505  020056  054450
011502  047440  020122  024516
011510  020077     100

011513     045  041442  046101  MES7:   .ASCII  '%"CALIBRATION TEST"%@'
011520  041111  040522  044524
011526  047117  052040  051505
011534  021124  040045
011540  021045  042522  047503  MES8:   .ASCII  '%"RECOVERY TEST"@'
011546  042526  054522  052040
011554  051505  021124     100

011561     045  044103  020056  MES9:   .ASCII  '%CH. @'
011566     100

011567     045  042447  054047  MES10:  .ASCII  "%'E'XT. OR 'I'NT. SYNC? @"
011574  027124  047440  020122
011602  044447  047047  027124
011610  051440  047131  037503
011616  040040


011620  051105  047522  020122  MES12:  .ASCII  'ERROR BIT SET!%@'
011626  044502  020124  042523
011634  020524  040045

011640  021045  042522  042520  MES13:  .ASCII  '%"REPEATIBILITY TEST"@'
011646  052101  041111  046111
011654  052111  020131  042524
011662  052123  040042

011666  022445  044103  024056  MES14:  .ASCII  '%%CH.(S)? @'
011674  024523  020077     100


011701     103  052517  052116  MES16:  .ASCII  'COUNT SPREAD? @'
011706  051440  051120  040505
```

```
     011714  037504  040040

     011720  040507  047111  051450  MES18:  .ASCII  'GAIN(S)? @'
     011726  037451  040040

     011732  020045  041440  027110  MES19:  .ASCII  '%  CH.   LO      AV      HI@'
     011740  020040  046040  020117
     011746  020040  020040  053101
     011754  020040  020040  044040
     011762  040111

     011764  020045  020040  047514  MES20:  .ASCII  '%   LO   -5   -4   -3   -2   -1   AV'
     011772  020040  026440  020065
     012000  020040  032055  020040
     012006  026440  020063  020040
     012014  031055  020040  026440
     012022  020061  020040  053101
     012030  025440  020061  020040          .ASCII' +1   +2   +3   +4   +5   HI@'
     012036  031053  020040  025440
     012044  020063  020040  032053
     012052  020040  025440  020065
     012060  020040  044510     100

     012065     045  047520  042527  MES21:  .ASCII  '%POWER FAILURE @'
     012072  020122  040506  046111
     012100  051125  020105     100

     012105     045  041445  027110  MES22:  .ASCII  '%%CH.? @'
     012112  020077     100

     012115     053  027065  030060  MES23:  .ASCII  '+5.00V@'
     012122  040126

     012124  053523  052111  044103  MES24:  .ASCII  'SWITCH NEG.!@'
     012132  047040  043505  020456
     012140     100

     012141     053  027062  030065  MES25:  .ASCII  '+2.50V@'
     012146  040126

     012150  030453  031056  053065  MES26:  .ASCII  '+1.25V@'
     012156     100

     012157     053  027060  031066  MES27:  .ASCII  '+0.625V@'
     012164  053065     100

     012167     053  027060  030463  MES28:  .ASCII  '+0.3125V@'
     012174  032462  040126

     012200  030053  030456  033065  MES29:  .ASCII  '+0.1563V@'
     012206  053063     100

     012211     053  027060  033460  MES30:  .ASCII  '+0.0781V@'
     012216  030470  040126

     012222  030053  030056  034463  MES31:  .ASCII  '+0.0390V@'
```

```
012230  053060     100

012233     045  040507  047111  MES32:  .ASCII  '%GAIN  5.0000  2.5000  1.2500  0.6250  '
012240  020040  027065  030060
012246  030060  020040  027062
012254  030065  030060  020040
012262  027061  032462  030060
012270  020040  027060  031066
012276  030065  020040
012302  027060  030463  032462          .ASCII  '0.3125  0.1563  0.0781  0.390@'
012310  020040  027060  032461
012316  031466  020040  027060
012324  033460  030470  020040
012332  027060  034463  040060

012340  043445  044501  020116  MES33:  .ASCII  '%GAIN  VOLTAGE AVERAGE%@'
012346  053040  046117  040524
012354  042507  040440  042526
012362  040522  042507  040045

012370  030445  020040  020040  MES34:  .ASCII  '%1     @'
012376  040040

012400  031045  020040  020040  MES35:  .ASCII  '%2     @'
012406  040040

012410  032045  020040  020040  MES36:  .ASCII  '%4     @'
012416  040040

012420  034045  020040  020040  MES37:  .ASCII  '%8     @'
012426  040040

012430  026455  026455  026455  MES38:  .ASCII  '------  @'
012436  020040     100
012441     045  054523  041516  MES39:  .ASCII  '%SYNC? @'
012446  020077     100
012451     045  046111  042514  MES40:  .ASCII  ;%ILLEGAL TRAP TO @;
012456  040507  020114  051124
012464  050101  052040  020117
012472     100
012473     040  051106  046517  MES41:  .ASCII  ; FROM @;
012500  040040
012502  044445  051516  043125  MES42:  .ASCII  '%INSUFFICIENT MEMORY@'
012510  044506  044503  047105
012516  020124  042515  047515
012524  054522     100
012527     045  047503  042522  MES43:  .ASCII  '%CORE MEMORY NOT INCREMENTED=@'
012534  046440  046505  051117
012542  020131  047516  020124
012550  047111  051103  046505
012556  047105  042524  036504
012564     100
012565     045  047111  051103  MES44:  .ASCII  '%INCREMENT MEMORY TEST@'
012572  046505  047105  020124
012600  042515  047515  054522
012606  052040  051505  040124
```

```
 2665  012614  047503  053116  051105  MES45:  .ASCII  /CONVERTIONS 'S'INGLE OR 'M'ULTIPLE  @/
 2666  012622  044524  047117  020123
 2667  012630  051447  044447  043516
 2668  012636  042514  047440  020122
 2669  012644  046447  052447  052114
 2670  012652  050111  042514  020040
 2671  012660     100
 2672  012661     045  047503  042522  MES46:  .ASCII  '%CORE LOC.INCREMENTED,# OF CONVERTIONS IN LOCATION%%@'
 2673  012666  046040  041517  044456
 2674  012674  041516  042522  042515
 2675  012702  052116  042105  021454
 2676  012710  047440  020106  047503
 2677  012716  053116  051105  044524
 2678  012724  047117  020123  047111
 2679  012732  046040  041517  052101
 2680  012740  047511  022516  040045
 2681  012746  046445  046505  051117  MES47:  .ASCII  '%MEMORY OFFSET = @'
 2682  012754  020131  043117  051506
 2683  012762  052105  036440  040040
 2684
 2685
```

```
  2686                                            .EVEN
  2687
  2688                                    ;ADDRESS AND CONSTANTS TABLE
  2689
  2690  012770  000000                    ADSIGN: 0                               ;UNIPOLAR=0, BIPOLAR=1
  2691  012772  000000                            SIGEXT: 0                       ;SIGN EXT .(INC MEM.)
  2692  012774  000000                    INTFLG: 0
  2693  012776  001000                    STACK:  1000                            ;INITIAL SP. ADDRESS
  2694  013000  001560                    AVECTR: INITB                           ;'↑A' VECTOR ADDRESS
  2695  013002  001722                    PVECTR: INIT3                           ;'↑P' VECTOR ADDRESS
  2696  013004  000000                    INCFLG: 0                               ;SOFTWARE FLAG;0=NO INC. MEM.
  2697  013006  000000                    MEMSIZ: 0                               ;CALCULATED MEM SIZE TO SUPPORT INC. MEM
  2698  013010  000000                    OFFSET:0                                ;CALCULATED MEMORY OFFSET.←←←←←←←
  2699  013012  000000                    SOFLAG: 0                               ;SOFTWARE 'FLAG'
  2700  013014  000000                    ADSIZE: 0                               ;OCTAL STORAGE OF A/D LENGTH
  2701  013016  000000                    PROC:   0                               ;TEMP STORAGE FOR 'PSW'
  2702  013020  000000                    CHRCNT: 0                               ;TEMP STORAGE
  2703  013022  000000                    COUNT:  0                               ;TEMP STORAGE
  2704  013024  000000                    ICOUNT: 0                               ;TEMP STORAGE
  2705  013026  000000                    HOLD:   0
  2706  013030  000000                    ITEST:  0
  2707  013032  000000                    STA:    0
  2708  013034  000000                    INITAL: 0                               ;PERMANENT STORAGE
  2709  013036  000000                    FINAL:  0                               ;PERMANENT STORAGE
  2710  013040  000000                    INITL2: 0                               ;PERMANENT STORAGE
  2711  013042  000000                    FINAL2: 0                               ;PERMANENT STORAGE
  2712  013044  000000                    KSTOR1: 0                               ;PERMANENT STORAGE
  2713  013046  000000                    KSTOR2: 0                               ;PERMANENT STORAGE
  2714  013050  000000                    KSTOR3: 0                               ;PERMANENT STORAGE
  2715  013052  000000                    KSTOR4: 0                               ;PERMANENT STORAGE
  2716  013054  000000                    TEMP1:  0                               ;TEMPORARY STORAGE
  2717  013056  000000                    TEMP2:  0                               ;TEMPORARY STORAGE
  2718  013060  000000                    TEMP3:  0                               ;TEMPORARY STORAGE
  2719  013062  000000                    SAVEPC: 0
  2720  013064  000000                    SAVPSW: 0
  2721  013066  000000                    SAV2PC: 0
  2722  013070  000000                    SAV2SW: 0
  2723  013072  000000                    RETSWH: 0
  2724  013074  000000                    MESPRT: 0
  2725  013076  000000                    HIGH:   0
  2726  013100  000000                    LOW:    0
  2727  013102  000000                    AVERM5: 0
  2728  013104  000000                    AVERM4: 0
  2729  013106  000000                    AVERM3: 0
  2730  013110  000000                    AVERM2: 0
  2731  013112  000000                    AVERM1: 0
  2732  013114  000000                    AVRAGE: 0
  2733  013116  000000                    AVERP1: 0
  2734  013120  000000                    AVERP2: 0
  2735  013122  000000                    AVERP3: 0
  2736  013124  000000                    AVERP4: 0
  2737  013126  000000                    AVERP5: 0
  2738  013130  000000                    ORLOW:  0
  2739  013132  000000                    MINUS5: 0
  2740  013134  000000                    MINUS4: 0
  2741  013136  000000                    MINUS3: 0
```

N05

```
  2742  013140  000000                  MINUS2: 0
  2743  013142  000000                  MINUS1: 0
  2744  013144  000000                  AVGCNT: 0
  2745  013146  000000                  PLUS1:  0
  2746  013150  000000                  PLUS2:  0
  2747  013152  000000                  PLUS3:  0
  2748  013154  000000                  PLUS4:  0
  2749  013156  000000                  PLUS5:  0
  2750  013160  000000                  ORHIGH: 0
  2751  013162  000000                  XSPRD1: 0
  2752  013164  000000                  XSPRD2: 0
  2753  013166  000000                  XSPRD3: 0
  2754  013170  000000                  XSPRD4: 0
  2755  013172  000000                  POS500: 0
  2756  013174  000000                  POS250: 0
  2757  013176  000000                  POS125: 0
  2758  013200  000000                  POS625: 0
  2759  013202  000000                  POS312: 0
  2760  013204  000000                  NEG500: 0
  2761  013206  000000                  NEG250: 0
  2762  013210  000000                  NEG125: 0
  2763  013212  000000                  NEG625: 0
  2764  013214  000000                  NEG312: 0
  2765  013216  000000                  GP50X1: 0
  2766  013220  000000                  GP25X1: 0
  2767  013222  000000                  GP12X1: 0
  2768  013224  000000                  GP62X1: 0
  2769  013226  000000                  GP31X1: 0
  2770  013230  000000                  GP25X2: 0
  2771  013232  000000                  GP12X2: 0
  2772  013234  000000                  GP62X2: 0
  2773  013236  000000                  GP31X2: 0
  2774  013240  000000                  GP15X2: 0
  2775  013242  000000                  GP12X4: 0
  2776  013244  000000                  GP62X4: 0
  2777  013246  000000                  GP31X4: 0
  2778  013250  000000                  GP15X4: 0
  2779  013252  000000                  GP07X4: 0
  2780  013254  000000                  GP62X8: 0
  2781  013256  000000                  GP31X8: 0
  2782  013260  000000                  GP15X8: 0
  2783  013262  000000                  GP07X8: 0
  2784  013264  000000                  GP03X8: 0
  2785  013266  000000                  GM50X1: 0
  2786  013270  000000                  GM25X1: 0
  2787  013272  000000                  GM12X1: 0
  2788  013274  000000                  GM62X1: 0
  2789  013276  000000                  GM31X1: 0
  2790  013300  000000                  GM25X2: 0
  2791  013302  000000                  GM12X2: 0
  2792  013304  000000                  GM62X2: 0
  2793  013306  000000                  GM31X2: 0
  2794  013310  000000                  GM15X2: 0
  2795  013312  000000                  GM12X4: 0
  2796  013314  000000                  GM62X4: 0
  2797  013316  000000                  GM31X4: 0
```

```
 013320  000000          GM15X4: 0
 013322  000000          GM07X4: 0
 013324  000000          GM62X8: 0
 013326  000000          GM31X8: 0
 013330  000000          GM15X8: 0
 013332  000000          GM07X8: 0
 013334  000000          GM03X8: 0
                         ;HERE STARTS A '30' WORD STATUS WORD BUFFER

 013336  000000          RANBUF: 0
         013434                  .=.+60.

                         ;HERE STARTS THE '512' WORD A/D DATA BUFFER.
 013434  000000          ADBUFF: 0

         001242                  .END    INIT
```

```
ADADR   001234        871#    1071*   1657*   1701*   1885*
ADBUFF  013434        1023    1031    1601    1887    1891    2220    2275    2811#
ADCNVT  006074        1020    1087    1510    1593    1659    1703    1884#
ADCR    001216        864#    1897*   1898*
ADCSR   001220        865#    1646*   1691*   1884*   1905    2430*   2437*
ADDBR   001226        868#
ADINT   001236        872#    2427*   2438*
ADLVL   001240        873#    2429*   2438    2439*
ADSIGN  012770        883*    913     1181    1207    1243    1291    1345    1393    1429    1453    1489    1818*   2349
                      2690#
ADSIZE  013014        932*    935     2224    2232    2243    2250    2254    2260    2264    2351    2700#
ADSWR   001222        866#    1660    1665    1672    1675    1886*
ADWCR   001224        867#    1018*   1085*   1509*   1592*   1658*   1702*   1888
ADWRA   001230        869#    1887*
ADWRB   001232        870#    1888*   1889*   1890*   1891*
ASTRIC  011455        2061    2508#
AVECTR  013000        958*    990*    1055*   1164*   1570*   1622*   1797    1954    2694#
AVERM1  013112        1518    2731#
AVERM2  013110        2730#
AVERM3  013106        2729#
AVERM4  013104        2728#
AVERM5  013102        2281    2284    2727#
AVERP1  013116        1516    2251    2733#
AVERP2  013120        2734#
AVERP3  013122        2735#
AVERP4  013124        2736#
AVERP5  013126        2279    2737#
AVGCNT  013144        1134    2301    2744#
AVGDAT  007702        2238#   2241
AVGTAB  007136        1023*   1466*   1601*   2097#   2098*
AVMEM   010352        2343#   2346
AVMEM1  010522        2375#   2379
AVRAGE  013114        1125    1512    1514    1556    2244*   2249    2259    2261    2348*   2364    2366*   2368*   2732#
BCDBIN  005710        1836#   1874    2125
BCDBN1  005730        1841#   1865
BCDBN2  005734        1843#   1861
BCDEND  006016        1844    1847    1862#
BCDERR  006054        1849    1851    1867    1869    1872#
BCDTAB  006064        889*    894*    1077    1839    1840*   1866    1868    1875#   1950    1951    1952    1953    1956
                      1968    2090
BINDEC= 104006        842#    1119    1140    1599    1733
CALBRT  001730        965     990#
CALBT1  001756        990     997#
CALBT2  002004        991     1004#   1039
CALBT3  002204        1033#   1035
CALBT4  002212        1026    1030    1036#
CALB1A  001762        996     999#
CALB2A  002120        1016    1018#   1038
CALB2B  002174        1022    1031#
CATOR1= 104005        841#    1093    1513
CATORZ  007726        2129    2248#
CATR1   010022        2271#   2273
CATR2   010042        2276#   2293    2295    2297
CATR3   010072        2285#   2289
CATR4   010110        2286    2291#
CATR5   010134        2277    2301#   2387
```

D06

```
ADF11A  PART II, ANALOG DIAGNOSTIC TEST MACY11 27(732)  26-OCT-76  16:42  PAGE 70
DZADHA.CMB      CROSS REFERENCE TABLE -- USER SYMBOLS

CHKCNT  002526          1102     1108#
CHRCNT  013020          1771*    1782*    1783     1810     1812*    1843     1845*    1864     2702#
CLRCOR  005326          1084     1647     1690     1744#
CLRCR1  005340          1747#    1749
CLRINT  010750           954     1904     2436#
CMPTE   007604          2128     2218#
CMPUTE= 104004           840#    1092     1511
CNTRLA  011445          1794     2502#
CNTRLC  011441           956     2500#
CNTRLP  011450          1802     2504#
CNVT1   006210          1893     1903#
CNVT2   006230          1906     1909#
CORSIZ  001410           914      916#
COUNT   013022          1476*    1481*    1486*    1581*    2147*    2703#
CRLF    011453          1003     1029     1111     1117     1137     1496     1525     1635     1683     1734     1736     1825     2506#
DASH6   007272          2137     2145#    2148
DCODER  007000          1804     2060#    2072     2076     2078     2086     2091
DECOCT= 104001           837#     886     1076     1945     1966     2089
DECOUT  007504          2188     2195#
DECPNT  007572          2180*    2184     2186     2193*    2210#
DECPRT  007404          2130     2178#
DEC1    007532          2196     2198     2201#
DEC2    007546          2200     2203#
DIGCNT  007566          2179*    2189*    2197     2208#
DIGIT   007564          2182*    2183*    2195     2199*    2202*    2205     2207#
DOT     011460           961     2510#
EMTOK   007216          2114     2116#
EMTSRV  007166           829     2110#
EMTTAB  007226          2118     2124#
ERTRAP  006576           821     2005#
EXCHAN  006446          1955     1957     1960#
EXCORE  005354          1745     1751#
EXGAIN  006510          1972     1975#
EXTTY   005656          1826#
FILE1   007746          2252#    2256
FILE2   010002          2262#    2266
FINAL   013036          1005*    1006*    1078     1080*    1082     1144*    1654*    1698*    1894*    1897     1946*    1950*    1958
                        2709#
FINAL2  013042          1146     1598     1948*    1952*    1958     2711#
GAIN    002706           971     1164#
GAINER  004124          1520#
GAININ= 104003           839#    1073     1580     2077
GANEXT  004134          1515     1517     1519     1522#
GDVAL   005010          1662     1669#
GERR1   004150          1521     1527#
GERR2   004200          1530     1534#
GERR3   004214          1535     1539#
GERR4   004232          1540     1544#
GERR5   004240          1545     1549#
GERR6   004254          1538     1543     1548     1551#
GETCHA= 104017           851#    1072     1169     1579     2075
GETDAT  007634          2225#    2236
GETREQ= 104021           853#    1796     1803     1826     1870     2163     2191     2415
GLOOP   004046          1508#    1528     1558
GMD3X8  013334          1462     2804#
GMD7X4  013322          1438     2799#
```

```
GM07X8   013332          1444    2803#
GM12X1   013272          1252    2787#
GM12X2   013302          1258    2791#
GM12X4   013312          1264    2795#
GM15X2   013310          1402    2794#
GM15X4   013320          1408    2798#
GM15X8   013330          1414    2802#
GM25X1   013270          1216    2786#
GM25X2   013300          1222    2790#
GM31X1   013276          1354    2789#
GM31X2   013306          1360    2793#
GM31X4   013316          1366    2797#
GM31X8   013326          1372    2801#
GM50X1   013266          1191    2785#
GM62X1   013274          1300    2788#
GM62X2   013304          1306    2792#
GM62X4   013314          1312    2796#
GM62X8   013324          1318    2800#
GP03X8   013264          1452    2784#
GP07X4   013252          1422    2779#
GP07X8   013262          1428    2783#
GP12X1   013222          1230    2767#
GP12X2   013232          1236    2771#
GP12X4   013242          1242    2775#
GP15X2   013240          1380    2774#
GP15X4   013250          1386    2778#
GP15X8   013260          1392    2782#
GP25X1   013220          1200    2766#
GP25X2   013230          1206    2770#
GP31X1   013226          1326    2769#
GP31X2   013236          1332    2773#
GP31X4   013246          1338    2777#
GP31X8   013256          1344    2781#
GP50X1   013216          1180    1466    2097    2765#
GP62X1   013224          1272    2768#
GP62X2   013234          1278    2772#
GP62X4   013244          1284    2776#
GP62X8   013254          1290    2780#
GT       002720          1164    1167#
GT0      002732          1168    1169#   1491    1494
GT1      002746          1175#
GT10     003654          1471#   1497
GT11     003746          1490    1492#
GT2      003004          1182    1195#
GT3      003062          1208    1225#
GT4      003160          1244    1267#
GT5      003276          1292    1321#
GT6      003414          1346    1375#
GT7      003512          1394    1417#
GT8      003570          1430    1447#
GT9      003626          1454    1466#
G1     = 010000           546#   1178    1189    1198    1214    1228    1250    1270    1298    1324    1352
G2     = 020000           547#   1204    1220    1234    1256    1276    1304    1330    1358    1378    1400    1539    1984
G4     = 040000           548#   1240    1262    1282    1310    1336    1364    1384    1406    1420    1436    1544    1989
G8     = 060000           549#   1288    1316    1342    1370    1390    1412    1426    1442    1450    1460    1994
HIGH     013076          1120    2222*   2226    2228*   2322*   2328*   2354    2356*   2358*   2725#
```

```
ADF11A  PART II, ANALOG DIAGNOSTIC TEST MACY11 27(732)  26-OCT-76  16:42  PAGE 72
DZADHA.CMB       CROSS REFERENCE TABLE -- USER SYMBOLS

HOLD    013026             1665*   1667    1675*   1677    1726*   1728    2705#
ICOUNT  013024             1004*   1470*   1573*   2095    2704#
INBUF   005670              963     966     969     972     975    1000    1067    1633    1638    1770    1772*   1831*   1838
                           2063    2073    2079    2085
INCFLG  013004              919*    927*   1062    1623    1744    2696#
INCTSA  004544             1622    1630#
INCTSB  004522             1623#
INCTST  004470              977    1618#
INIT    001242              832     879#   2813
INITA   001476              917     920#
INITAL  013034             1007*   1008*   1009*   1017*   1059*   1070*   1079*   1081*   1082*   1143*   1167*   1170*   1500*
                           1501*   1574*   1577*   1584    1656*   1700*   1895    1898    1947*   1951*   1967    2708#
INITB   001560              943#    958    2694
INITL2  013040             1575*   1578*   1580    1949*   1953*   1973    2710#
INIT1   001260              883#    900     941
INIT2   001604              945     958#    980
INIT3   001722              959     978#   2695
INPUTA  005426             1773#   1774    1788    1811
INPUTB  005460             1781#   1821
INTFLG  012774             2692#
INTVC  =%000004             552#
ITEST   013030             1638*   1639    1641    2706#
KSTOR1  013044             1010*   1011*   1012*   1013*   1032    1078*   1080    1083    1467*   1492*   1502*   1534    1539
                           1544    2712#
KSTOR2  013046             1014*   1037    1506*   1512*   2713#
KSTOR3  013050             1001*   1025*   1027*   1077*   1100    1645*   1679*   1681*   2090*   2714#
KSTOR4  013052             1083*   1118    1145*   1146    2715#
LDINTR  010714             1892    2427#
LDNEG   001366              909#    912
LDPOS   001350              903#    906
LDSIZE  001342              890     895     901#
LOW     013100             1122    2223*   2229    2231*   2323*   2333*   2359    2361*   2363*   2726#
MASK    010700             2400*   2406    2412*   2417#
MEMSIZ  013006              930*   1722    1748    1750*   2338    2381    2697#
MESPRT  013074             1060*   1112    1129    1133*   1171*   1529    1531*   2724#
MES10   011567              995    1631    2531#
MES12   011620             1908    2538#
MES13   011640             1057    2542#
MES14   011666             1944    2547#
MES16   011701             1075    2098    2551#
MES18   011720             1965    2555#
MES19   011732             1115    2559#
MES2    011105              885    2456#
MES20   011764             1132    2564#
MES21   012065             1938    2576#
MES22   012105             2580#
MES23   012115             1176    2583#
MES24   012124             1187    1212    1248    1296    1350    1398    1434    1458    2586#
MES25   012141             1196    2590#
MES26   012150             1226    2593#
MES27   012157             1268    2596#
MES28   012167             1322    2599#
MES29   012200             1376    2602#
MES3    011123             1166    2460#
MES30   012211             1418    2605#
MES31   012222             1448    2608#
```

```
MES32    012233              1469     2611#
MES33    012310              1533     2624#
MES34    012370              1472     1537     2629#
MES35    012400              1475     1542     2632#
MES36    012410              1480     1547     2635#
MES37    012420              1485     1550     2638#
MES38    012430              2146     2641#
MES39    012441               998     2083     2643#
MES4     011264               944     2479#
MES40    012451              2009     2645#
MES41    012473              2014     2649#
MES42    012502              1626     2651#
MES43    012527              1664     2655#
MES44    012565              1629     2661#
MES45    012614              1636     2665#
MES46    012661              1716     2672#
MES47    012746              1713     2681#
MES6     011466              1065     2515#
MES7     011513               993     2520#
MES8     011540              1572     2524#
MES9     011561              1597     2528#
MINUS1   013142              2303     2743#
MINUS2   013140              2306     2742#
MINUS3   013136              2309     2741#
MINUS4   013134              2312     2740#
MINUS5   013132              2292*    2374     2739#
MONITR   001566               833      952#    1627     1807     1939     2018
NEG125   013210              1251     1305     1365     1413     2762#
NEG250   013206              1215     1257     1311     1371     2761#
NEG312   013214               911     1353     1401     1437     1461     2764#
NEG500   013204               905     1190     1221     1263     1317     2760#
NEG625   013212              1299     1359     1407     1443     2763#
NOGAIN   005572              1993     1997#
NOP    = 000240               559#
NXADDR   010332              2327     2337#
OCTPRT   010576              2132     2396#
OFFSET   013010               920*     944*    1071     1657     1701     1715     1718     1746     1885     2318     2372     2375     2698#
ORHIGH   013160              2272     2294*    2370*    2383     2388     2750#
ORLOW    013130              1138     2270     2296*    2369*    2738#
OUTPTA   005476              1785#    1786     1815     1819
OVRHI    010120              2290     2294#
OVRLO    010126              2292     2296#
PARMC    007042              2065     2073#
PARMCS   007110              2074     2087#
PARMG    007054              2067     2077#
PARMS    007060              2069     2079#
PC     =%000007               542#     954*    1020*    1084*    1087*    1510*    1593*    1647*    1659*    1690*    1703*    1751*    1892*
                             1904*    1909*    1969*    1974*    1980*    1985*    1990*    1995*    2188*    2206*    2432*    2440*
PLUS1    013146              2302     2745#
PLUS2    013150              2305     2746#
PLUS3    013152              2308     2747#
PLUS4    013154              2311     2748#
PLUS5    013156              2749#
POPR0  = 012600               556#
POP1SP=  005726               554#    1997
POP2SP=  022626               558#    1795     1800     1903     2006
```

```
ADF11A  PART II, ANALOG DIAGNOSTIC TEST MACY11 27(732)  26-OCT-76  16:42  PAGE 74
DZADHA.CMB      CROSS REFERENCE TABLE -- USER SYMBOLS

POS125  013176      1229    1277    1337    1391    2757#
POS250  013174      1199    1235    1283    1343    2756#
POS312  013202      1325    1379    1421    1451    2759#
POS500  013172       901    1179    1205    1241    1289    2755#
POS625  013200      1271    1331    1385    1427    2758#
PRINT = 104000       836#    881     884     898     940     943     955     960     978     992     994     997    1002
                    1028    1056    1064    1074    1110    1114    1116    1131    1136    1165    1175    1186    1195
                    1211    1225    1247    1267    1295    1321    1349    1375    1397    1417    1433    1447    1457
                    1468    1471    1474    1479    1484    1495    1524    1532    1536    1541    1546    1549    1571
                    1596    1625    1628    1630    1634    1636    1643    1663    1682    1713    1716    1734    1736
                    1793    1801    1824    1828    1872    1907    1937    1943    1964    2008    2013    2060    2070
                    2082    2087    2145
PROC    013016      1000*   1015    1633*   1651    1695    1920*   1926    2085*   2701#
PRTADR  004256      1504*   1514    1516    1518    1552#
PRTAVG= 104014       848#   1024    1473    1478    1483    1488    1602
PRTOCT= 104010       844#   1121    1124    1127    1551    1555    1666    1676    1714    1727    2011    2016    2096
PSW     001200       857#    880*   1618*   1925*   2431*   2436*
PUSHR0= 010046       555#
PUSH1S= 005746       553#
PUSH2S= 024646       557#
PVECTR  013002       959*    991*   1061*   1168*   1576*   1621*   2081    2695#
PWRFAL  006232       823    1913#
PWRUP   006266      1921    1925#
QMARK   011463       899     940     979    1643    1629    1873    2071    2512#
QUANT   005072      1642    1690#   1738
RANBUF  013336      1582    1886    2807#
ROMEM = 104002       838#   1090
ROMEM1  010274      2325#   2341
ROMEM2  010276      2326#   2332    2335
RECVRY  004276       974    1570#
RECVY1  004310      1570    1573#
RECVY2  004354      1576    1581#
RECVY3  004372      1584#   1586
RECVY4  004406      1588#   1590
RECVY5  004416      1591#   1595    1603
REPTST  002226       968    1055#
REPT1   002240      1055    1058#   1954
REPT1A  002330      1063    1068    1072#
REPT2   002350      1061    1078#
REPT2A  002364      1080#   1148
REPT3   002414      1084#   1147
REPT3A  002446      1089    1092#
REPT3B  002452      1091    1094#
REPT4   002534      1095    1107    1110#
REPT5   002552      1113    1116#
REPT6   002620      1130    1133#
REPT6A  002646      1139#   1142
REPT7   002660      1097    1109    1135    1143#
RETA    004634      1621    1646#   1668    1674    1684
RETSWH  013072      1499*   1522    1557*   2723#
R0      =%000000     535#    901*    903*    905     909*    911     934*    935     938*   1031*   1098*   1101    1103*
                    1672*   1653*   1660    1913    1933*   2219*   2234*   2238*   2275*   2278    2318*   2325    2338
                    2240    2397*   2402*   2403*   2404*   2405
R1      =%000001     536#    887*    892*    915*    916*    921     923*    925*    926     930     933*    937*    942
                    1032*   1034*   1099*   1104    1105    1108    1582*   1584*   1588*   1718*   1720    1722    1724
                    1726    1732    1746*   1747*   1748    1775*   1776*   1777    1779    1781    1787    1791    1798
```

```
                         1805   1808   1813*  1816   1820   1822   1841*  1854   1855*  1856*  1857*  1858*  1859*
                         1860*  1863   1914   1932*  1967*  1973*  1977*  1984*  1989*  1994*  2027   2050*  2063*
                         2064   2066   2068   2218*  2235*  2237*  2240*  2248*  2255*  2258*  2265*  2283*  2287*
                         2288   2291*  2292*  2319*  2328   2329   2333   2337*  2374*  2377*  2380*  2383   2388
                         2390*  2399*  2413*
R2       =%000002         537#   888*   893*   902    907    931*   932   1100*  1101   1118*  1139*  1583*  1585*
                         1587*  1589*  1590*  1732*  1842*  1854*  1858   1859   1915   1931*  1968*  1970   1971
                         1978   1982   1987   1992   2028   2049*  2157*  2161   2165   2167   2173   2184*  2186*
                         2225*  2226   2228   2229   2231   2232*  2233   2249*  2250*  2252*  2253   2259*  2260*
                         2262*  2263   2284*  2285   2320*  2330*  2343*  2405*  2406*  2407*  2411
R3       =%000003         538#   891*   896*   902*   903    904*   907*   908*   909    910*  1839*  1863*  1916
                         1930*  2029   2048*  2221*  2222   2223   2224*  2233*  2239*  2242*  2243*  2244   2251*
                         2253*  2254*  2261*  2263*  2264*  2270*  2271*  2272   2274*  2276*  2321*  2329*  2344*
                         2347*  2348   2371*  2372*  2373*  2375   2378   2380   2381   2385
R4       =%000004         539#  1138*  1139   1141   1770*  1781*  1814   1838*  1846   1848   1850   1852*  1853
                         1862   1917   1929*  2030   2047*  2220*  2221   2225   2278*  2279   2281   2285   2325*
                         2326*  2342*  2345*  2351*  2353*  2354   2358   2359   2363   2364   2368
R5       =%000005         540#  1918   1920*  1934*  1935*  2031   2046*
SAVEPC    013062         2023*  2035   2042*  2054   2719#
SAVPSW    013064         2024*  2034   2043*  2053   2720#
SAVREG=   104020          852#  1769   1836   2156   2178   2396
SAV2PC    013066         2025*  2033   2044*  2052   2721#
SAV2SW    013070         2026*  2032   2045*  2051   2722#
SET       010536         2376   2380#  2386
SHIFT     010624         2402#  2414
SIGEXT    012772          942*  2356   2361   2366   2691#
SIXDSH=   104015          849#  1477   1482   1487
SIZE      001316          892#   897
SOFLAG    013012         1058*  1069*  1088   1620*  1711   1717*  2324*  2331   2334*  2699#
SP       =%000006         541#   879*   953*  1501   1502   1503*  1504   1505*  1506   1507*  1619*  1913*  1914*
                         1915*  1916*  1917*  1918*  1919*  1920   1926*  1927   1928   1929   1930   1931   1932
                         1933   2005   2007   2023   2024   2025   2026   2027*  2028*  2029*  2030*  2031*  2032*
                         2033*  2034*  2035*  2042   2043   2044   2045   2046   2047   2048   2049   2050   2051*
                         2052*  2053*  2054*  2110*  2111*  2112*  2113   2116*  2117*  2118*  2119*  2120   2157
                         2158*  2397   2398*  2427   2428*
SPACE  =  104007          843#  1120   1123   1126   1553   1554   1600   1678   1730   2100
SPACEX    005406         1729*  1761*  1763*  1765#  2099*
SPCHR     005512         1778   1780   1791#
SPCHRP    005534         1792   1798#
SPCHR1    005556         1799   1805#
SPCHR2    005620         1809   1816#
SPCHR3    005636         1817   1820#
SPCHR4    005644         1822#
SPCHR5    005662         1784   1823   1828#
STA       013032         1640*  1649*  1650*  1653*  1654   1655*  1656   1660*  1661   1692*  1693*  1694*  1697*
                         1698   1699*  1700   2707#
STACK     012776          879    953   1619   2693#
SWR       001212          862#  1005   1007   1010   1014   1021   1037   1094   1096   1520   1527   1594   1648
                         1670   1692
SWR0      001214          863#
SW00   =  000001          531#
SW01   =  000002          530#
SW02   =  000004          529#
SW03   =  000010          528#
SW04   =  000020          527#
SW05   =  000040          526#
```

```
SW06  = 000100          525#
SW07  = 000200          524#
SW08  = 000400          523#
SW09  = 001000          522#
SW10  = 002000          521#
SW11  = 004000          520#
SW12  = 010000          519#    1094
SW13  = 020000          518#    1096    1527    1594
SW14  = 040000          517#    1520
SW15  = 100000          516#
TAGA    007662         2230     2232#
TAKEGN= 104013          847#    1203    1219    1233    1239    1255    1261    1275    1281    1287    1303    1309    1315
                       1329     1335    1341    1357    1363    1369    1383    1389    1405    1411    1425    1441
TEMP1   013054         2005*    2010*   2012    2716#
TEMP2   013056         2007*    2015*   2017    2717#
TEMP3   013060         2095*    2101*   2718#
TITLE   011001          882     2443#
TKB     001204          859#    1775
TKS     001202          858#    1773    2421
TKSFLG  010702         2138     2421#
TPB     001210          861#    1760*   1787*   2167*   2169*   2172*   2205*   2411*
TPS     001206          860#    1758    1785    2159    2170    2203    2409
TSLO    007650         2227     2229#
TSTCNT  002506         1101#    1106
TSTCT4  002462         1096#
TSTTKS= 104016          850#    1019    1036    1096    1508    1591    1669    1719    1731    1737    2174    2187    2408
TTYIN = 104011          845#     962     999    1066    1498    1632    1637    1837    2062    2084    2423
TYPECL  007354         2166     2169#
TYPERA  007320         2159#    2160    2168    2175
TYPER1  007340         2162     2165#
TYPER2  007346         2167#
TYPMES  007306         2124     2156#
TYPT1   007430         2182#    2194
TYPT2   007436         2183#    2185
TYPT3   007474         2190     2193#
WAITGN= 104012          846#    1177    1188    1197    1213    1227    1249    1269    1297    1323    1351    1377    1399
                       1419     1435    1449    1459
XCHAIN  006336         1943#    1959    2139
XGAINA  006450         1964#    1998    2127
XGAINB  006512         1969     1974    1977#
XGAIN2  006526         1979     1982#
XGAIN4  006542         1983     1987#
XGAIN8  006556         1988     1992#
XGETRG  006724         2042#    2141
XPRTAV  007126         2095#    2136
XPTA1   007134         2096#    2102
XROMEM  010246         2126     2318#
XSAVRG  006650         2023#    2140
XSPACE  [illegible]    1758#    1762    2131
XSPRO1  013162         1099     1141    2301*   2302*   2303*   2304    2751#
XSPRO2  013164         2304*    2305*   2306*   2307    2752#
XSPRO3  013166         2307*    2308*   2309*   2310    2753#
XSPRO4  013170         1105     2310*   2311*   2312*   2754#
XTTYIN  005410          826     1769#   1830    2133
XWATGN  003766         1498#    2134    2135
ZERO    007570         2181*    2199    2201*   2209#
```

```
.      = 013436              562#    563     565     567     569     571     573     575     577     579     581     583     585
                             587     589     591     593     595     597     599     601     603     605     607     609     611
                             613     615     617     619     621     623     625     627     629     631     633     635     637
                             639     641     643     645     647     649     651     653     655     657     659     661     663
                             665     667     669     671     673     675     677     679     681     683     685     687     689
                             691     693     695     697     699     701     703     705     707     709     711     713     715
                             717     719     721     723     725     727     729     731     733     735     737     739     741
                             743     745     747     749     751     753     755     757     759     761     763     765     767
                             769     771     773     775     777     779     781     783     785     787     789     791     793
                             795     797     799     801     803     805     807     809     811     813     815     817     820#
                             825#    828#    831#    856#    922     964     967     970     973     976    1493    1523    1652
                            1671    1680    1696    1759    1806    1832#   1896    1936    2080    2171    2204    2210    2401
                            2410    2422    2808#
```

```
COMMEN        18
ENDCOM        18
ESCAPE        18
GETPRI        18
GETSWR        18
MULT          18
NEWTST        18
POP           18
PUSH          18
REPORT        18
SETPRI        18
SETUP         18
SKIP          18
SLASH         18
STARS         18
SWRSU         18
TYPBIN        18
TYPDEC        18
TYPNAM        18
TYPNUM        18
TYPOCS        18
TYPOCT        18
TYPTXT        18
$$ESCA        18
$$NEWT        18
$$SKIP        18
.EQUAT        18
.HEADE        18
.KT11         18
.SETUP        18
.SWRHI        18
.$ACT1        18
.$APTB        18
.$APTH        18
.$APTY        18
.$ASTA        18
.$CATC        18
.$CMTA        18
.$DB2D        18
.$DB2O        18
.$DIV         18
.$EOP         18
.$ERRO        18
.$ERRT        18
.$MULT        18
.$POWE        18
.$RAND        18
.$RDDE        18
.$RDOC        18
.$READ        18
.$R2AZ        18
.$SAVE        18
.$SB2D        18
.$SB2O        18
.$SCOP        18
.$SIZE        18
```

M06

```
.$SUPR       1#
.$TRAP       1#
.$TYPB       1#
.$TYPD       1#
.$TYPE       1#
.$TYPO       1#
.$4OCA       1#
.1170        1#
```

N06

```
ADF11A  PART II, ANALOG DIAGNOSTIC TEST MACY11 27(732)  26-OCT-76  16:42  PAGE 82
DZADHA.CMB      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADC     2234  2242  2330  2347
ADD      923   925  1503  1505  1507  1858  1859  1860  1891  2098  2118  2158  2186  2193  2224
        2232  2233  2250  2260  2302  2303  2305  2306  2308  2309  2311  2312  2329  2356  2361
        2366  2372  2398  2428
ASL      892   893   915   931   937   938  1855  1856  1857  1890  2116  2291  2353  2371
ASR      904   910  1012  2238  2343
BCC      939
BEQ      890   895   914   922  1038  1063  1089  1102  1109  1135  1182  1208  1244  1292  1346
        1394  1430  1454  1515  1517  1519  1521  1523  1640  1642  1662  1723  1745  1811  1821
        1847  1957  1972  2065  2067  2069  2080  2166  2198  2277  2286  2339  2350  2382  2384
        2389
BGE     1147  2376
BGT     1762  1851  2230
BIC     1006  1008  1013  1081  1500  1649  1655  1693  1699  1776  1852  1947  1949  1977  2117
        2358  2363  2368  2437
BICB    2406
BIS      916  1009  1017  1070  1079  1082  1133  1170  1501  1531  1577  1578  1650  1653  1694
        1697  1884  1894  1951  1953  1984  1989  1994  2202  2407
BIT      921   935  1094  1096  1520  1527  1594  2354  2359  2364
BITB    2409
BLE     1844  2227
BLO     1959
BLT     1849
BMI     1671  1712  1778  1780  1784  2280  2282  2327
BNE      906   912   936   964   967   970   973   976  1016  1026  1068  1095  1097  1106  1113
        1130  1142  1490  1493  1528  1530  1535  1540  1545  1586  1595  1624  1652  1680  1696
        1721  1749  1792  1799  1806  1809  1817  1823  1865  1936  1955  1979  1983  1988  1993
        2074  2102  2114  2148  2162  2190  2196  2236  2241  2256  2266  2273  2289  2332  2346
        2355  2360  2365  2414
BPL      897  1022  1035  1590  1759  1774  1786  1867  1869  1896  1906  2160  2171  2185  2204
        2410  2422
BR       900   941   945   980   996  1030  1039  1091  1107  1148  1497  1538  1543  1548  1558
        1603  1644  1725  1735  1788  1815  1819  1830  1861  1998  2072  2076  2078  2086  2091
        2168  2175  2194  2200  2293  2295  2297  2335  2341  2357  2362  2367  2379  2386  2391
        2401
CLC     2352
CLR      883   919   929  1058  1059  1060  1167  1171  1499  1574  1575  1620  1747  1750  1763
        1771  1772  1840  1841  1842  1934  1946  1948  2219  2271  2283  2319  2320  2321  2322
        2323  2324  2369  2370  2377  2390  2431  2439
CMP      905   911   972   975  1015  1037  1067  1101  1105  1108  1134  1141  1146  1514  1516
        1518  1539  1544  1639  1641  1651  1661  1695  1722  1748  1783  1798  1850  1866  1868
        1954  1958  1978  1982  1987  1992  2064  2066  2068  2197  2226  2229  2272  2279  2281
        2285  2288  2338  2375  2381  2383  2388
CMPB     963   966   969  1777  1779  1791  1805  1808  1816  1820  1822  1846  1848  2161  2165
DEC      894   896  1025  1034  1492  1585  1589  1679  1717  1761  1812  1845  2101  2147  2235
        2240  2255  2262  2265  2276  2326  2334  2345  2413
EMT      836   837   838   839   840   841   842   843   844   845   846   847   848   849   850
         851   852   853
HALT    1922  2115  2290
INC     1103  1143  1144  1145  1557  1782  1935  2183  2189  2252  2287  2292  2294  2296  2337
JMP      832   833   965   968   971   974   977  1491  1494  1627  1668  1674  1684  1738  1797
        1804  1807  1874  1939  2018  2081  2120  2387
JSR      954  1020  1084  1387  1510  1593  1647  1659  1690  1703  1892  1904  1969  1974  2188
MOV      879   880   887   888   891   901   902   903   907   909   917   918   920   924   927
         928   930   932   933   934   942   953   958   959   990   991  1000  1001  1004  1005
        1007  1010  1014  1018  1023  1027  1031  1032  1055  1061  1069  1071  1077  1079  1080
```

```
            1083    1085    1098    1099    1100    1118    1138    1139    1164    1168    1466    1467    1470    1476    1481
            1486    1502    1504    1506    1509    1512    1570    1573    1576    1581    1582    1583    1584    1587    1588
            1592    1598    1601    1618    1619    1621    1622    1633    1638    1645    1646    1648    1654    1656    1657
            1658    1660    1665    1672    1675    1681    1691    1692    1698    1700    1701    1702    1718    1726    1729
            1732    1746    1760    1770    1775    1781    1813    1818    1838    1839    1853    1854    1863    1885    1886
            1887    1888    1897    1898    1913    1914    1915    1916    1917    1918    1919    1920    1921    1925    1926
            1927    1928    1929    1930    1931    1932    1933    1950    1952    1967    1968    1973    2005    2007    2023
            2024    2025    2026    2027    2028    2029    2030    2031    2032    2033    2034    2035    2042    2043    2044
            2045    2046    2047    2048    2049    2050    2051    2052    2053    2054    2063    2085    2090    2095    2099
            2110    2112    2119    2157    2169    2172    2179    2180    2181    2182    2199    2201    2205    2218    2220
            2221    2222    2223    2225    2228    2231    2237    2244    2248    2249    2251    2253    2258    2259    2261
            2263    2270    2274    2275    2278    2284    2301    2304    2307    2310    2318    2325    2328    2333    2342
            2348    2351    2374    2380    2397    2399    2400    2412    2427    2429    2430    2436    2438
MOVB        1787    2167    2405    2411
NEG          908    1889
RESET        952    1033    1673
ROL         2402    2403    2404
ROR         2239    2344
RTI         1526    1764    1827    1871    1960    1975    2036    2055    2103    2149    2164    2192    2245    2313    2416
            2424
RTS         1751    1909    1980    1985    1990    1995    2206    2432    2440
SUB          889    2010    2015    2111    2184    2243    2254    2264    2373
SWAB        1011
TST          913     926    1021    1062    1088    1104    1112    1129    1181    1207    1243    1291    1345    1393    1429
            1453    1489    1522    1529    1534    1623    1670    1711    1720    1724    1744    1810    1814    1843    1862
            1864    1895    1905    1956    1970    1971    2073    2079    2113    2195    2331    2340    2349    2378    2385
TSTB        1758    1773    1785    2159    2170    2173    2203    2421
WAIT        1899
.ABS         508
.ASCII      2443    2451    2456    2460    2470    2479    2486    2495    2500    2502    2504    2506    2508    2510    2512
            2515    2520    2524    2528    2531    2538    2542    2547    2551    2555    2558    2564    2570    2576    2580
            2583    2586    2590    2593    2596    2599    2602    2605    2608    2611    2618    2624    2629    2632    2635
            2638    2641    2643    2645    2649    2651    2655    2661    2665    2672    2681
.BYTE       2442
.ENABL         1
.END        2813
.EVEN       2686
.LIST          1
.MACRO         1
.NLIST         1
.REM           1
.REPT        563
.TITLE       507


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 #,DZADHA.SEQ=SYSMAC.CO,DZADHA.CMB
 RUN-TIME: 26 35 4 SECONDS
 RUN-TIME RATIO: 258/65=3.9
 CORE USED:  33K  (65 PAGES)
```

C07

```
Spooler runtime 10 Seconds, 44 MCS, 257 disk reads, 6 disk writes, 79 pages
```

[redacted]

```
00000001111111111222222222233333333334444444444555555555566666666667777777777888888888899999999990000000000111111111122222222223331
```

```
  10                    ...B1    2160  007324  100375  ...B5
  34                    ...C1    2216                  ...C5
                        ...D1    2255  007756  005301  ...D5
  91                    ...E1    2307  010200  016767  ...E5
                        ...F1    2363  010446  040467  ...F5
 151                    ...G1    2419                  ...G5
                        ...H1    2450  011052  027463  ...H5
 211                    ...I1    2506  011453     045  ...I5
                        ...J1    2562  011762  040111  ...J5
 271                    ...K1    2618  012302  027060  ...K5
                        ...L1    2674  012674  041516  ...L5
 329                    ...M1    2695  013002  001722  ...M5
                        ...N1    2751  013162  000000  ...N5

 389                    ...B2    2807  013336  000000  ...B6
 440                    ...C2                          ...C6
 459                    ...D2  CNTRLP  011450          ...D6
 482                    ...E2  GM31X1  013276          ...E6
 538          000003    ...F2  INITA   001476          ...F6
 569  000014  000016    ...G2  MES40   012451          ...G6
 625  000174  000176    ...H2                          ...H6
 681  000354  000356    ...I2                          ...I6
 737  000534  000536    ...J2  SW15  = 100000          ...J6
 793  000714  000716    ...K2                          ...K6
 843          104007    ...L2  REPORT       1#  CROSS RE ...L6
 883  001260  005067    ...M2                          ...M6
 937  001540  006301    ...N2          1847    1957    ...N6

 990  001730  012767    ...B3          2110    2112    ...B7
                        ...C3   **END** USER DAVIES,TOM ...C7
1049                    ...D3
1098  002472  012700    ...E3
1125  002576  013114    ...F3
1158                    ...G3
1211  003036  104000    ...H3
1263  003154  013204    ...I3
1317  003272  013204    ...J3
1371  003410  013206    ...K3
1425  003526  104013    ...L3
1472  003656  012370    ...M3
1528  004156  001333    ...N3

1570  004276  012767    ...B4
1613                    ...C4
1669  005010  104016    ...D4
1725  005256  000771    ...E4
1751  005354  000207    ...F4
1798  005534  022701    ...G4
1842  005732  005002    ...H4
1888  006124  017777    ...I4
1932  006314  012601    ...J4
1971  006474  005712    ...K4
2027  006670  010146    ...L4
2065  007016  001411    ...M4
2113  007200  005716    ...N4
```