ADF11

ANALOG TESTS

MD-11-DZADH-A

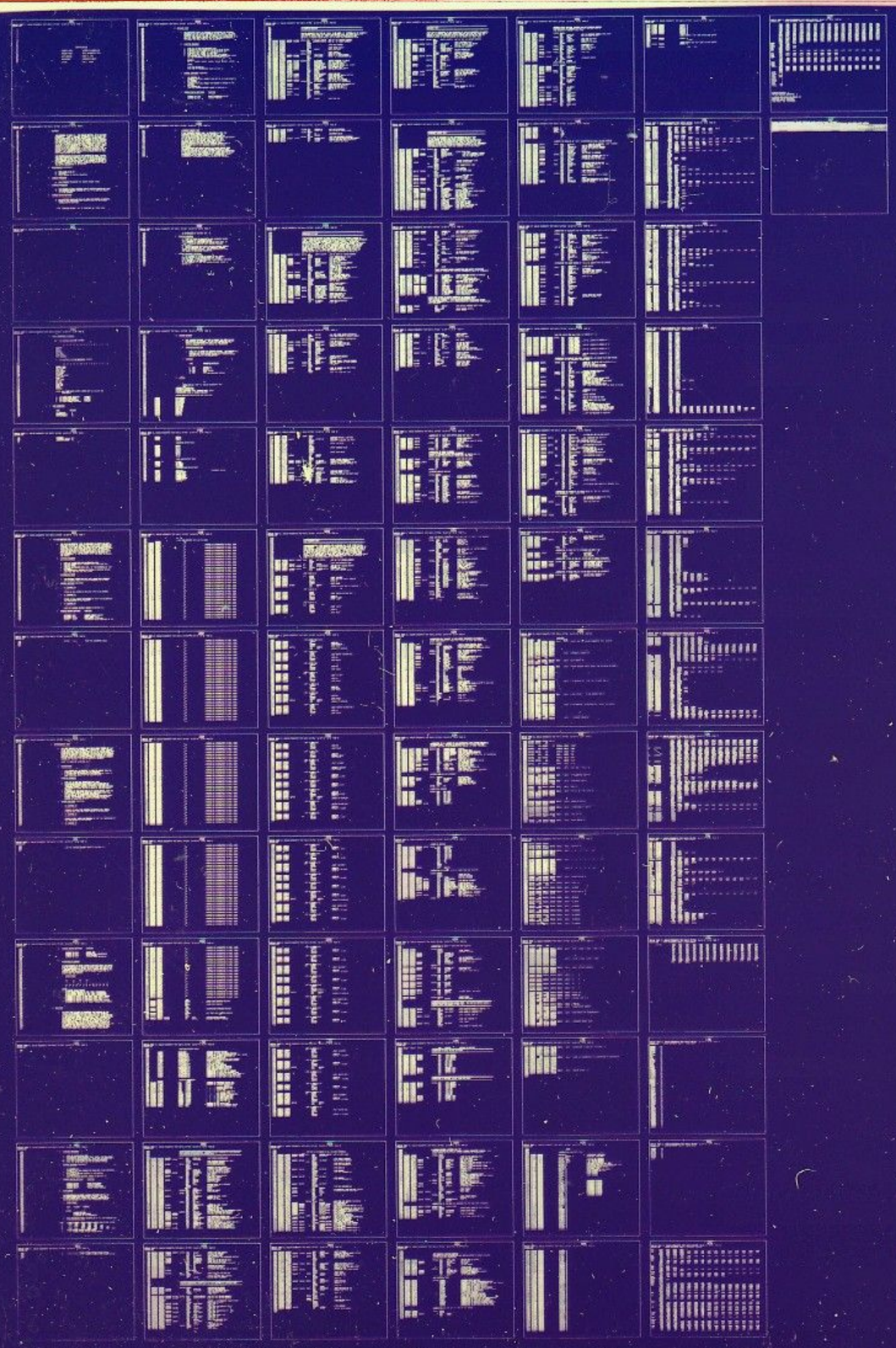
EP-DZADH-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A



```
                                .REM    X
```

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-D[illegible]08-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 SENTENCE IS THEN TYPED GIVING THE LETTER DESIGNATORS TO BE TYPED TO ENTER ONE OF THE FOUR (4) SEPARATE 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 USING A '^C' (OBTAINED VIA TYPING THE 'CNTR' AND 'C' KEYS SIMULTANEOUSLY WHILE RUNNING ANY TEST WILL ENABLE THE PROGRAM TO THE KEYBOARD MONITOR AND AWAITS A NEW LETTER 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 & NOXM
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-
```

B. 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'

```
                                        SWITCH   '15=1'        PRINT THE CONVERTED VALUE
199
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,M(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 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 INDICATION OF ALL '1024' COUNTS WHEN USING THE 'INCREMENTALLY' FEATURE SINCE NO ATTEMPT IS MADE TO CATEGORIZE 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 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 'AOF11'. 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	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 ADDDRESS IN THE (SWR) 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 PROGRAM TO RETURN TO THE MONITOR.
3. ↑P (CONTROL P) LOAD THE COMMAND DECODER (REFER TO SECTION 13).

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


```
                X
                .TITLE ADF11A  PART II, ANALOG DIAGNOSTIC TEST
                .ABS
;MAINDEC-11-DZRDH-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=X0
                000001          R1=X1
                000002          R2=X2
                000003          R3=X3
                000004          R4=X4
                000005          R5=X5
                000006          SP=X6
                000007          PC=X7

                                ;GAIN EQUIVALENCE TABLE

                000000          G1=000
                020000          G2=20000
                040000          G4=40000
                060000          G8=60000
                                ;INSTRUCTIONS DEFINITIONS

                000004          INTVC=X4                        ;INTERRUPT VECTOR
                025746          PUSH1SP=5746
                005726          POP1SP=5726
                010046          PUSH1R0=10046
                012600          POP1R0=12600
                025746          PUSH2SP=25746
                022626          POP2SP=22626
                000240          NOP=240
```

```
                                ;LOAD TRAP CATCHER INTO LOC'S 0-1000
000000 000000                           .=0
000000 000002                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000002 000004                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000004 000006                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000006 000010                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000010 000012                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000012 000014                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000014 000016                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000016 000020                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000020 000022                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000022 000024                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000024 000026                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000026 000030                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000030 000032                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000032 000034                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000034 000036                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000036 000040                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000040 000042                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000042 000044                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000044 000046                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000046 000050                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000050 000052                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000052 000054                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000054 000056                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000056 000060                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000060 000062                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000062 000064                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000064 000066                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000066 000070                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000070 000072                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000072 000074                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000074 000076                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000076 000100                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000100 000102                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000102 000104                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000104 000106                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000106 000110                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000110 000112                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000112 000114                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000114 000116                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000116 000120                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000120 000122                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000122 000124                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000124 000126                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000126 000130                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000130 000132                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000132 000134                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000134 000136                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000136 000140                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000140 000142                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000142 000144                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000144 000146                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000146 000150                           4$:+2                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
000150 000152                           .+2
```

```
 516  000152  000004                    4                   ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 517  000154  000156                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 518  000156  000002                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 519  000160  000162                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 520  000162  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 521  000164  000166                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 522  000166  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 523  000170  000172                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 524  000172  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 525  000174  000176                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 526  000176  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 527  000200  000202                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 528  000202  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 529  000204  000206                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 530  000206  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 531  000210  000212                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 532  000212  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 533  000214  000216                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 534  000216  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 535  000220  000222                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 536  000222  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 537  000224  000226                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 538  000226  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 539  000230  000232                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 540  000232  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 541  000234  000236                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 542  000236  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 543  000240  000242                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 544  000242  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 545  000244  000246                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 546  000246  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 547  000250  000252                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 548  000252  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 549  000254  000256                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 550  000256  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 551  000260  000262                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 552  000262  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 553  000264  000266                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 554  000266  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 555  000270  000272                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 556  000272  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 557  000274  000276                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 558  000276  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 559  000300  000302                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 560  000302  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 561  000304  000306                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 562  000306  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 563  000310  000312                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 564  000312  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 565  000314  000316                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 566  000316  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 567  000320  000322                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 568  000322  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 569  000324  000326                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 570  000326  000004                    4.+2                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
 571  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          ROMEM=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          GETCHR=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    SWR0:   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    ADADR:  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. VALUES?
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 @ 1000 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  010[illegible]67  011272                  MOV     R2,ADSIZE       ;SAVE IT
0[illegible]1522  012701  176000                  MOV     #176000,R1      ;BASE VAL OF AD EXT OF SIGN
0[illegible]15[illegible]6  012700  002000                  MOV     #2000,R0        ;BASE VAL OF A/D SIZE
0[illegible]1[illegible]32  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
001[illegible]52  00[illegible]42                          BR      INIT1
001[illegible]54  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  0167[illegible]6  011202                  MOV     STACK,SP        ;RESET STACK POINTER
001574  00[illegible]767  007150                  JSR     PC,CLRINT       ;CLR A/D INTERRUPT VECTOR
001600  1040[illegible]0                          PRINT                   ;CALL MESSAGE PRINTER
001602  011441                          CNTRLC                  ;TYPE '↑C'

001604  012767  001560  0[illegible]6  INIT2:  MOV     #INITB,AVECTR   ;SET UP '↑A' VECTOR ADDRESS.
0[illegible]612  012[illegible]57  001722  011162                  MOV     #INIT3,PVECTR   ;SET UP '↑P' VECTOR ADDRESS
0[illegible]620  1040[illegible]0                          PRINT
[illegible]2  0114[illegible]0                          DOT                     ;PRINT '.' TO INDICATE MONITOR READY
[illegible]4  104011                          TTYIN                   ;WAIT FOR TTY ENTRY
[illegible]626  122767  000103  004034          CMPB    #103,INBUF      ;TEST FOR 'C'
[illegible]4  [illegible]1002                          BNE     .+6             ;NOT 'C'
[illegible]  [illegible]167  000066                  JMP     CALBRT          ;YES, RUN 'CALIBRATION' TEST
[illegible]  1[illegible]2767  000122  004020          CMPB    #122,INBUF      ;TEST FOR 'R'
[illegible]  [illegible]1002                          BNE     .+6             ;NOT 'N'
[illegible]  [illegible]167  000350                  JMP     REPTST          ;YES, RUN 'REPEATIBILITY' TEST
[illegible]  [illegible]2767  000107  004004          CMPB    #107,INBUF      ;TEST FOR 'G'
[illegible]664  0[illegible]1002                          BNE     .+6             ;NOT 'G'
[illegible]66  [illegible]167  001014                  JMP     GAIN            ;YES, RUN 'GAIN' TEST
[illegible]2  [illegible]2767  000105  003770          CMP     #105,INBUF      ;TEST FOR 'E'
0[illegible]700  001002                          BNE     .+6             ;NOT 'E'
0[illegible]702  0[illegible]167  002370                  JMP     RECVRY          ;YES, RUN RECOVERY TEST
0[illegible]706  0[illegible]2767  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
0017[illegible]4  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 '1A' RESTART ADDRESS
001736  012767  002004  011036           MOV     #CALBT2,PVECTR  ;SET UP '1P' START ADDRESS.
001744  104000                           PRINT
001746  011513                           MFS7                    ;TYPE TEST HEADER
001750  104400                           PRINT
001752  011557                           MES10                   ;TEXT 'SYNC I OR E'
001754  000432                           ER      CALB1A          ;WAIT FOR INPUT
001756  104400                  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
                                        PRINT
                                        CRLF
                        011453
                        012767  000001  011012  CALBT2: MOV     #1,ICOUNT       ;SETUP TO PRINT '1' VALUE
                        017767  177174  011016           MOV     @SWR,FINAL      ;GET CH. FROM SW. REG.
                        042767  177000  011010           BIC     #177000,FINAL   ;CLR UNWANTED BITS
                        017767  177160  011000           MOV     @SWR,INITA      ;ALSO GET AS INITIAL CH. & GAIN
                        042767  107000  010772           BIC     #107000,INITAL  ;CLR UNWANTED BITS
                        052767  110000  010764           BIS     #110000,INITAL  ;SELECT SEQ. DMA, INITAL
                        017767  177136  010766           MOV     @SWR,KSTOR1     ;GET DELAY BITS 9-11
                        000367  010762                   SWAB    KSTOR1
                        006267  010756                   ASR     KSTOR1
                        042767  177770  010750           BIC     #177770,KSTOR1  ;DELAY NOW SET
                        017767  177112  010744           MOV     @SWR,KSTOR2     ;SAVE ORIGINAL SWITCH SETTING.
                        022767  000105  010706           CMP     #105,PROC       ;TEST SYNC SELECT
                        001003                           BNE     CALB2A          ;BRANCH IF NOT 'E'
                        052767  004000  010714           BIS     #4000,INITAL    ;SET 'EXT' SYNC ENABLE
                        012777  177777  177076  CALB2A:  MOV     #-1,@ADWCR      ;SET UP FOR '1' CONVERSION
                        104016                           TSTTKS
                        004767  003740                   JSR     PC,ADCNVT       ;TAKE AND STORE THE CONVERSIONS
                        005777  177052                   TST     @SWR            ;TEST FOR SW15 TO PRINT
                        100015                           BPL     CALB2B          ;BRANCH IF NOT SET.
                        012767  013434  004766           MOV     #AVGBUFF,AVGTAB ;SET UP TO PRINT VALUE
                        104014                           PRTAVG                  ;PRINT IT
                        005367  010672                   DEC     KSTOR3
                        001015                           BNE     CALBT4
                        012767  000011  010662           MOV     #11,KSTOR3
                        104000                           PRINT
                        011453                           CRLF
                        000407                           BR      CALBT4          ;TEST FOR LOOP
```

```
 1031  002174  016700  011234          CALB2B: MOV     ADBUFF,R0       ;SET UP A/D BUFFER.
 1032  002200  016701  010640                  MOV     KSTOR1,R1       ;SET UP DELAY (RESET COUNT)
 1033  002204  000005                  CALBT3: RESET
 1034  002206  005301                          DEC     R1              ;DECREMENT DELAY
 1035  002210  100375                          BPL     CALBT3
 1036  002212  104016                  CALBT4: TSTTKS                  ;TEST FOR KEYBOARD INTERRUPT
 1037  002214  026777  010626  176770          CMP     KSTOR2,@SWR     ;TEST IF SWITCH REGISTER HAS CHANGED
 1038  002222  001736                          BEQ     CALB2A          ;BRANCH AND TAKE NEXT CONVERSION
 1039  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  010630                  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     REPTIA          ;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     REPTIA          ;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,@ADRCR   ;LOAD THE OFFSET REG.++++++++++++
002330  104017                  REPTIA: 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  010412  010436          MOV     KSTOR1,KSTOR4   ;SAVE STARTING CH;
002414  004767  002706          REPT3:  JSR     PC,CLRCOR       ;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?
```

```
 1089  002440  001402                          BEQ     REPT3A          ;NO, EBA (EARL BOUSE ARITHMETIC)
 1090  002442  104002                          RDMEM                   ;YES, READ MEMORY & COLLECT VALUES
 1091  002444  000432                          BR      REPT3B
 1092  002446  104004                  REPT3A: CMPUTE                  ;AVERAGE & COMPUTE DISTRIBUTION
 1093  002450  104005                          CATORIZ
 1094  002452  032777  010000  176532  REPT3B: BIT     @SW12,@SWR      ;TEST DATA SW12
 1095  002460  001025                          BNE     REPT4           ;IF SET, FORCE TYPE OUT
 1096  002462  032777  020000  176522  TSTCT4: BIT     @SW13,@SWR      ;TEST FOR INHIBIT TYPEOUT
 1097  002470  001073                          BNE     REPT7           ;BRANCH IF SW SET
 1098  002472  012700  000001                  MOV     #1,R0
 1099  002476  012701  013162                  MOV     #XSPR01,R1
 1100  002502  016702  010342                  MOV     KSTOR3,R2
 1101  002506  020002                  TSTCNT: CMP     R0,R2
 1102  002510  001406                          BEQ     CHKCNT
 1103  002512  005200                          INC     R0
 1104  002514  005721                          TST     (R1)+           ;UPDATE COUNT ADDRESS
 1105  002516  020127  013172                  CMP     R1,#XSPRD4+2    ;CHECKED '1-4'?
 1106  002522  001371                          BNE     TSTCNT          ;NO
 1107  002524  000403                          BR      REPT4           ;ILLEGAL ENTRY, TYPE OUT COUNTS
 1108  002526  022711  002000          CHKCNT: CMP     #2000,(R1)      ;ARE ALL COUNTS IN COUNT SPREAD?
 1109  002532  001452                          BEQ     REPT7           ;YES, EXIT
 1110  002534  104030                  REPT4:  PRINT
 1111  002536  011453                          CRLF
 1112  002540  005767  010330                  TST     MESPRT          ;TEST IF HEADER HAS BEEN TYPED
 1113  002544  001002                          BNE     REPT5           ;BRANCH IF YES
 1114  002546  104000                          PRINT
 1115  002550  011732                          MES19                   ;TEXT 'CH. HIGH AVG. LOW'
```

```
                                        REPT5:  PRINT
                                                CRLF                    ;CARRIAGE RETURN, LINE FEED
                010270                          MOV     KSTOR4,R2       ;MOV. CH.
                                                BINDEC                  ;CONVERT TO DECIMAL AND PRINT
                                                SPACE
                                                PRTOCT                  ;PRINT LOW VALUE
                                                LOW
                                                SPACE
                                                PRTOCT                  ;PRINT AVERAGE VALUE
                                                AVRAGE
                                                SPACE
                                                PRTOCT                  ;PRINT HIGH VALUE
                                                HIGH
                010262                          TST     MESPRT
                                                BNE     REPT6
                                                PRINT
                                                MES20                   ;PRINT 'COUNT SPREAD' HEADER
                000007  010246  REPT6:          BIS     #7,MESPRT       ;INHIBIT OTHER HEADERS
                002000  010310                  CMP     #2000,AVGCNT    ;TEST IF ALL COUNTS WERE AT AVG.
                                                BEQ     REPT7           ;BRANCH TO NEXT CH. IF YES.
                                                PRINT
                                                CRLF
                013130                          MOV     #ORLOW,R4
                                        REPT2:  MOV     (R4)+,R2
                                                TSTEC                   ;TYPE OUT COUNT SPREAD
                013162                          CMP     #XSPRD1,R4      ;TEST FOR DONE
                                                BEQ     REPT6A          ;BRANCH IF NO AND TYPE NEXT COUNT
                010150          REPT7:          INITAL
                010146                          FINAL
                010156                          KSTOR4                  ;INCREMENT 'CH.'
                010142  010150                  CMPB    FINAL2,KSTOR4   ;TESTED ALL CH.(S)?
                                                BGE     REPT3           ;BRANCH IF NO AND TEST NEXT CH.
                                                BR      REPT2A          ;OTHERWISE RESET AND REPEAT
```

```
1149
1150                                    ;*************************************************************************
1151                                    ;GAIN ACCURACY TEST
1152                                    ;*************************************************************************
1153
1154                                    ;THE GAIN ACCURACY TEST REQUESTS FOR A DECIMAL CH. TO BE TYPED IN VIA
1155                                    ;THE OPERATOR.  IT THEN REQUESTS '16' SPECIFIED VOLTAGES TO BE APPLIED TO
1156                                    ;THAT CHANNEL.  AFTER SUPPLY THE REQUESTED VOLTAGE, THE OPERATOR TYPES A
1157                                    ;SPACE ON THE TELETYPE AND A SERIES OF '1024' CONVERSIONS A E THEN TAKEN
1158                                    ;AT SPECIFIED GAIN SETTINGS AND AVERAGED OUT.  IF THE AVERAGED VALUE
1159                                    ;IS FOUND TO BE MORE THAN + OR -1 COUNT FROM THE INPUT VOLTAGES TRUE
1160                                    ;VALUE, IT IS CONSIDERED IN ERROR AND THE ERROR VOLTAGE, THE EXPECTED VALUE
1161                                    ;AND THE GAIN ARE TYPED OUT.  AFTER COMPUTING INPUT VOLTAGE AVERAGES, A
1162                                    ;TABLE OF ALL CONVERSION RESULTS ARE TYPED OUT.
1163
1164  002706  012767  002720  010064  GAIN:   MOV     #GT,AVECTR      ;SET UP '1A' RETURN ADDRESS
1165  002714  104000                    PRINT
1166  002716  011123                    MES3                    ;TEXT GAIN ACCURACY TEST
1167  002720  005067  010110  GT:      CLR     INITAL
1168  002724  012767  002734  010050  MOV     #GTD+2,PVECTR   ;SET UP '1P' VECTOR ADDRESS
1169  002732  104017            GTD:    GETCHA                  ;REQUEST CH.
1170  002734  052767  110000  010072  BIS     #110000,INITAL  ;SEQ., DMA, INITAL
1171  002742  005067  010126          CLR     MESPRT          ;CLR PRINT INHIBIT SWITCH
1172
1173                                    ;TEST +5.0V X G1
1174
1175  002746  104000            GT1:    PRINT
1176  002750  012115                    MES23                   ;TEXT '+5.00V'
1177  002752  104012                    WAITGN                  ;CALL THE GAIN 'WAIT' HANDLER
1178  002754  000000                    G1                      ;GAIN X1
1179  002756  013172                    POSSDD
1180  002760  013216                    GPSDX1                  ;SAVE VALUE
1181  002762  005767  010002          TST     ADSIGN
1182  002766  001406                    BEQ     GT2             ;BRANCH TO NEXT TEST IF UNIPOLAR.
1183
1184                                    ;TEST -5.0V X G1
1185
1186  002770  104000                    PRINT
1187  002772  012124                    MES24                   ;TEXT 'SWITCH VOLTAGE NEG.'
1188  002774  104012                    WAITGN
1189  002776  000000                    G1                      ;GAIN X1
1190  003000  013204                    NEGSDD                  ;SHOULD=-5.0V
1191  003002  013266                    GMSDX1                  ;SAVE VALUE
1192
1193                                    ;TEST +2.5V X G1
1194
1195  003004  104000            GT2:    PRINT
1196  003006  012141                    MES25                   ;TEXT '+2.5V'
1197  003010  104012                    WAITGN
1198  003012  000000                    G1                      ;GAIN X1
1199  003014  013174                    PDSSD
1200  003016  013220                    GPSDX1                  ;SAVE VALUE
1201
```

```
1202
1203  003030  104013                ;TEXT +2.5V X G2 (5.0V)
                                        TAKEGN
1204  003032  020000                   G2                      ;GAIN X2
1205  003034  013172                   POS500                  ;SHOULD=+5.0V
1206  003036  013230                   GP2SX2                  ;SAVE VALUE
1207  003040  005767  007734           TST     ADSIGN
1208  003044  001412                   BEQ     GT3             ;BRANCH IF UNIPOLAR
1209
1210                                  ;TEST -2.5V X G1
1211  003046  104000                   PRINT
1212  003050  012124                   MES24                   ;TEXT SWITCH VOLTAGE NEG.
      003052  104012                   WAITGN
      003054  000000                   G1                      ;GAIN X1
      003056  013210                   NEG250
1216  003060  013270                   GM2SX1                  ;SAVE VALUE
1217
1218                                  ;TEST -2.5V X G2
1219  003062  104013                   TAKEGN
1220  003064  020000                   G2                      ;GAIN X2
1221  003066  013204                   NEG500
1222  003070  013300                   GM2SX2                  ;SAVE VALUE
1223
1224                                  ;TEST +1.25V X G1
1225  003072  104000                GT3:    PRINT
1226  003074  012150                   MES26                   ;TEXT '+1.25V'
1227  003076  104012                   WAITGN
1228  003100  000000                   G1                      ;GAIN X1
1229  003102  013126                   POS125
1230  003104  013222                   GP12X1                  ;SAVE VALUE
1231
1232                                  ;TEST '+1.25V X G2
1233  003106  104013                   TAKEGN
1234  003110  020000                   G2                      ;GAIN X2
1235  003112  013174                   POS250                  ;=TO +2.5V
1236  003114  013232                   GP12X2                  ;SAVE VALUE
1237
1238                                  ;TEST '+1.25 X G4
1239  003116  104013                   TAKEGN
1240  003120  040000                   G4                      ;GAIN X4
1241  003122  013172                   POS500
1242  003124  013[illegible]            GP12X4                  ;SAVE VALUE
1243  003126  007767  007646           TST     ADSIGN
1244  003132  001416                   BEQ     GT4             ;BRANCH IF UNIPOLAR
1245
1246                                  ;TEST '-1.25V X G1
1247  003134  104000                   PRINT
1248  003136  012124                   MES24                   ;TEXT 'SWITCH VOLTAGE NEG.'
1249  003140  104012                   WAITGN
1250  003142  000000                   G1                      ;GAIN X1
1251  003144  013210                   NEG125
1252  003146  013272                   GM12X1                  ;SAVE VALUE
1253
```

```
1254
1255  003140  104013                ;TEST -1.25V X G2
                                      TAKEGN
1256  003142  020020                 G2                      ;GAIN X2
1257  003144  013106                 NEG250
1258  003146  013302                 GM12X2                  ;SAVE VALUE
1259
1260                                ;TEST -1.25V X G4
1261  003150  104013                 TAKEGN
1262  003152  040000                 G4                      ;GAIN X4
1263  003154  013204                 NEG500                  ;SHOULD = 5.0V
1264  003156  013312                 GM12X4                  ;SAVE VALUE
1265
1266                                ;TEST +0.625V X G1
1267  003160  104000                GT4:  PRINT
1268  003162  012157                 MES27                   ;TEXT '+0.625V'
1269  003164  104012                 WAITGN
1270  003166  000000                 G1
1271  003170  013100                 POS625                  ;SHOULD = +0.625V
1272  003172  013224                 GP62X1                  ;SAVE VALUE
1273
1274                                ;TEST +0.625V X G2
1275  003174  104013                 TAKEGN
1276  003176  020000                 G2                      ;GAIN X2
1277  003200  013176                 POS125                  ;SHOULD = +1.25V
1278  003202  013234                 GP62X2                  ;SAVE IT
1279
1280                                ;TEST +0.625V X G4
1281  003204  104013                 TAKEGN
1282  003206  040000                 G4                      ;GAIN X4
1283  003210  013174                 POS250                  ;SHOULD = +2.5V
1284  003212  013244                 GP62X4                  ;SAVE IT
1285
1286                                ;TEST +0.625V X G8
1287  003214  104013                 TAKEGN
1288  003216  060000                 G8                      ;GAIN X8
1289  003220  013172                 POS500                  ;SHOULD = +5.00V
1290  003222  013254                 GP62X8                  ;SAVE IT
1291  003224  005767  007540        TST     ADSIGN
1292  003230  001422                 BEQ     GT5             ;BRANCH IF UNIPOLAR
1293
1294                                ;TEST -0.625V X G1
1295  003232  104000                PRINT
1296  003234  012124                 MES24                   ;SWITCH VOLTAGE NEG.
1297  003236  104012                 WAITGN
1298  003240  000000                 G1                      ;GAIN X1
1299  003242  013212                 NEG625                  ;SHOULD = -0.625V
1300  003244  013274                 GM62X1
1301
1302                                ;TEST -0.625V X G2
1303  003246  104013                 TAKEGN
1304  003250  020000                 G2                      ;GAIN X2
1305  003252  013210                 NEG125                  ;SHOULD = -1.25V
1306  003254  013304                 GM62X2                  ;SAVE IT
1307
```

```
                                        ;TEST -0.625V X G4
                                                TAKEGN
                                                G4                      ;GAIN X4
                                                NEG250                  ;SHOULD = -2.5V
                                                GM62X4

                                        ;TEST -0.625V X G8
                                                TAKEGN
                                                G8                      ;GAIN X8
                                                NEG500                  ;SHOULD = -5.00V
                                                GM62X8

                                        ;TEST +0.3125V X G1
                                        GT5:    PRINT
                                                MES28                   ;TEXT '+0.3125V'
                                                WAITGN
                                                G1                      ;GAIN X1
                                                POS312                  ;SHOULD = +0.3125V
                                                GP31X1                  ;SAVE IT

                                        ;TEST +0.3125V X G2
                                                TAKEGN
                                                G2                      ;GAIN X2
                                                POS625                  ;SHOULD = +0.625V
                                                GP31X2

                                        ;TEST +0.3125V X G4
                                                TAKEGN
                                                G4                      ;GAIN X4
                                                POS125                  ;SHOULD = +1.25V
                                                GP31X4

                                        ;TEST +0.3125V X G8
                                                TAKEGN
                                                G8                      ;GAIN X8
                                                POS250                  ;SHOULD = +2.50V
                                                GP31X8
                                                TST     POSIGN
                                                BEQ     GT6             ;BRANCH IS UNIPOLAR

                                        ;TEST -0.3125V X G1
                                                PRINT
                                                MES24                   ;TEXT 'SWITCH NEG.'
                                                WAITGN
                                                G1                      ;GAIN X1
                                                NEG312                  ;SHOULD = -0.3125V
                                                GM31X1

                                        ;TEST -0.3125V X G2
                                                TAKEGN
                                                G2                      ;GAIN X2
                                                NEG625                  ;SHOULD = -0.625V
                                                GM31X2
```

```
1362
1363  003374  104013                ;TEST -0.3125V X G4
                                            TAKEGN
1364  003376  040000                      G4                    ;GAIN X4
1365  003400  013210                      NEG125                ;SHOULD = -1.25V
1366  003402  013316                      GM31X4
1367
1368                                ;TEST -0.3125V X G8
1369  003404  104013                      TAKEGN
1370  003406  060000                      G8                    ;GAIN X8
1371  003410  013206                      NEG250                ;SHOULD = -2.50V
1372  003412  013326                      GM31X8
1373
1374                                ;TEST +0.1563V X G2
1375  003414  104000                GT6:  PRINT
1376  003416  012200                      MES29                 ;TEXT '+0.1563V'
1377  003420  104012                      WAITGN
1378  003422  020000                      G2                    ;GAIN X2
1379  003424  013232                      POS312                ;SHOULD = +0.3125V
1380  003426  013240                      GP15X2
1381
1382                                ;TEST +0.1563V X G4
1383  003430  104013                      TAKEGN
1384  003432  040000                      G4                    ;GAIN X4
1385  003434  013200                      POS625                ;SHOULD = +0.625V
1386  003436  013250                      GP15X4
1387
1388                                ;TEST +0.1563V X G8
1389  003440  104013                      TAKEGN
1390  003442  060000                      G8                    ;GAIN X8
1391  003444  013176                      POS125                ;SHOULD = +1.25V
1392  003446  013260                      GP15X8
1393  003450  005767  007314              TST     ADSIGN
1394  003454  001416                      BEQ     GT7           ;BRANCH IF UNIPOLAR
1395
1396                                ;TEST -0.1563V X G2
1397  003456  104000                      PRINT
1398  003460  012124                      MES24                 ;TEXT 'SWITCH NEG.
1399  003462  104012                      WAITGN
1400  003464  020000                      G2                    ;GAIN 'X2'
1401  003466  013214                      NEG312                ;SHOULD = -0.3125V
1402  003470  013310                      GM15X2
1403
1404                                ;TEST -0.1563V X G4
1405  003472  104013                      TAKEGN
1406  003474  040000                      G4                    ;GAIN X4
1407  003476  013212                      NEG625                ;SHOULD = -0.625V
1408  003500  013320                      GM15X4
1409
1410                                ;TEST -0.1563V X G8
1411  003502  104013                      TAKEGN
1412  003504  060000                      G8                    ;GAIN X8
1413  003506  013210                      NEG125                ;SHOULD = -1.25V
1414  003510  013330                      GM15X8
1415
```

```
                                        ;TEST +0.0781V X G4
                                GT7:    PRINT
                                        MES30                   ;TEXT '+0.0781V'
                                        WAITGN
                                        G4                      ;GAIN X4
                                        POS312                  ;SHOULD = +0.3125V
                                        GP07X4

                                        ;TEST +0.0781V X 8
                                        TAKEGN
                                        G8                      ;GAIN X8
                                        POS625                  ;SHOULD = +0.625V
                                        GP07X8
                                        TST     ADSIGN
                                        BEQ     GT8

                                        ;TEST -0.0781V X G4
                                        PRINT
                                        MES24                   ;TEXT 'SWITCH NEG.'
                                        WAITGN
                                        G4                      ;GAIN X4
                                        NEG312                  ;SHOULD = -0.3125V
                                        GM07X4

                                        ;TEST -0.0781V X G8
                                        TAKEGN
                                        G8                      ;GAIN X8
                                        NEG625                  ;SHOULD = -0.625V
                                        GM07X8

                                        ;TEST +0.0390V X G8
                                GT8:    PRINT
                                        MES31                   ;TEXT '+0.0390V'
                                        WAITGN
                                        G8                      ;GAIN X8
                                        POS312                  ;SHOULD = +0.3125V
                                        GP03X8
                                        TST     ADSIGN
                                        BEQ     GT9

                                        ;TEST -0.0390V X G8
                                        PRINT
                                        MES24                   ;TEXT 'SWITCH NEG.'
                                        WAITGN
                                        G8
                                        NEG312                  ;SHOULD = -0.3125V
                                        GM03X8
```

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

003626 012767 013216 003302 GT9:    MOV     #GPSOX1,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     GTO             ;OTHERWISE RESTART GAIN TEST
003746 005367 007072        GT11:   DEC     KSTOR1
003752 001002                       BNE     .+6
003754 000167 176752                JMP     GTO
003760 104000                       PRINT
003762 011453                       CRLF
003764 000733                       ER      GT10
003766 104011              XWATGN:  TTYIN                   ;WAIT FOR 'CR' BEFORE CONTINUING
003770 005067 007076                CLR     RETSW#          ;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 OF 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
004052 012777 176000 175146         MOV     #-2000,@ADWCR   ;SET UP TO TAKE '1024' CONVERSIONS
004060 004767 002012                JSR     PC,ADCNVT       ;TAKE THE CONVERSIONS
004064 104004                       CMPUTE                  ;COMPUTE THE AVERAGE
004066 016777 007024 006F7A         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     GAINEXT         ;EXIT IF EQUAL
004104 026777 007006 000144         CMP     AVERP1,@PRTADR  ;IS AVERAGE =TO KNOWN VALUE +1?
004112 001410                       BEQ     GAINEXT         ;EXIT IF EQUAL
004114 027767 000136 006770         CMP     @PRTADR,AVERM1  ;IS AVERAGE =TO KNOWN VALUE -1?
```

```
1519  004122  001404                          BEQ     GAINEXT         ;EXIT IF EQUAL
1520  004124  032777  040000  175060  GAINER: BIT     #SW14,@SWR      ;TEST FOR INHIBIT SCOPE LOOPING
1521  004132  001406                          BEQ     GERR1           ;BRANCH IF NOT SET
1522  004134  005767  006732          GAINEXT: TST    RETSWH          ;WAS AN ERROR REPORTED?
1523  004140  001402                          BEQ     .+6             ;NO, CONTINUE
1524  004142  104000                          PRINT
1525  004144  011453                          CRLF
1526  004146  000002                          RTI
1527  004150  032777  020000  175034  GERR1: BIT     #SW13,@SWR      ;TEST FOR PRINT INHIBIT
1528  004156  001333                          BNE     GLOOP           ;BRANCH IS SW SET
1529  004160  005767  006710                  TST     MESPRT          ;TEST IF TITLE HAS BEEN TYPED
1530  004164  001005                          BNE     GERR2           ;BRANCH IF YES
1531  004166  052767  000001  006700          BIS     #1,MESPRT       ;OTHERWISE TYPE ERROR HEADER
1532  004174  104000                          PRINT
1533  004176  012340                          MES33                   ;TEXT 'GAIN
1534  004200  005767  006640          GERR2: TST     KSTOR1          ;TEST FOR 'G1'
1535  004204  001003                          BNE     GERR3           ;BRANCH IF NOT
1536  004206  104000                          PRINT
1537  004210  012350                          MES34                   ;TEXT '1'
1538  004212  000430                          BR      GERR6
1539  004214  022767  020000  006622  GERR3: CMP     #G2,KSTOR1      ;TEST FOR GX2
1540  004222  001003                          BNE     GERR4           ;BRANCH IF NOT -
1541  004224  104000                          PRINT
1542  004226  012400                          MES35                   ;TEXT '2'
1543  004230  000411                          BR      GERR6
1544  004232  022767  040000  006604  GERR4: CMP     #G4,KSTOR1      ;TEST FOR GAIN OF '4'
1545  004240  001003                          BNE     GERR5           ;BRANCH AND PRINT GX8
1546  004242  104000                          PRINT
1547  004244  012410                          MES36                   ;TEXT '4'
1548  004246  000402                          BR      GERR6
1549  004250  104000          GERR5:  PRINT
1550  004252  012420                          MES37                   ;TEXT '8'
1551  004254  104010          GERR6:  PRTOCT
1552  004256  000000          PRTAGR: 0                               ;TYPE VOLTAGE VALUE
1553  004260  104007                          SPACE                   ;TYPE SPACE
1554  004262  104007                          SPACE
1555  004264  104010                          PRTOCT
1556  004266  013114                          AVRAGE                  ;TYPE AVERAGE
1557  004270  005267  006576                  INC     RETSWH
1558  004274  000664                          BR      GLOOP           ;RETEST GAIN AVERAGE
1559
1560
```

```
                                        ;*****************************************************************
                                        ;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 '1A' RETURN ADDRESS
004304  104000                          PRINT
004306  011540                          .+58                    ;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                          ME59                    ;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
```

```
1604
1605
1606                                    ;*****************************************************************
1607                                    ;INCREMENT MEMORY TEST
1608                                    ;*****************************************************************
1609
1610                                    ;THIS TEST IS DESIGNED TO TAKE CONVERTIONS ONE AT A TIME
1611                                    ;IN THE SINGLE CHANNEL MODE USING INCREMENT MEMORY. THE
1612                                    ;SWR REGISTER IS READ AFTER EACH CONVERTION AND THE SPECIFIED
1613                                    ;ADDRESS IS TESTED FOR MODIFICATION. THEM THE CONTENTS OF THE
1614                                    ;SWR IS PRINTED OR DISPLAYED.IF SWITCH REG. BIT15=1 THE INCREMENTED
1615                                    ;ADDRESS IS DISPLAYED THROUGH R0 OTHERWISE PRINTED ON THE TELETYPE
1616                                    ;THE OPERATOR MAY CHECK THAT EACH ADDRESS LINE IS SELECTABLE
1617
1618  004470  012777  000340  174502  INCTST: MOV     #340,@PSW
1619  004476  016706  001274          MOV     STACK,SP
1620  004502  005067  001304          CLR     SWFLAG          ;CLEAR PRINT FLAG
1621  004506  012767  004534  006266  MOV     #.+ETA,PVECTR           ;SET UP A CONTROL P START
1622  004514  012767  004544  006256  MOV     #INCTSA,AVECTR          ;SET UP A RESTART ADDRESS
1623  004522  005767  006256          INCTSB: TST     INCFLG          ;TEST IF CORE AVAIL
1624  004526  001404                  BEQ     1$
1625  004530  104401                  PRINT
1626  004532  012742                  MFS#2                   ; REPORT INSUFFICIENT CORE
1627  004536  000167  175026          JMP     MONITR          ; RETURN TO MONITOR
1628  004542  104400          1$:     PRINT
1629  004544  012765                  MES#4                   ;TEXT INSUFFICIENT CORE
1630  004550  104400          INCTSA: PRINT
1631  004552  011567                  MES10                   ;SYNC I OR E
1632  004556  104411                  TTYIN                   ; 1 CHARACTER
1633  004560  016767  001112  006236  MOV     INBUF,PROC      ;SAVE IN TEMP STORE
1634  004566  104400                  PRINT
1635  004570  011453                  CRLF
1636  004572  104400  012614          PRINT,MES45                     ;ASK FOR TEST
1637  004576  104411                  TTYIN                   ;RECIEVE 1 CHAR.
1638  004600  016767  001072  006230  MOV     INBUF,ITEST     ;SAVE TTY CHAR
1639  004606  020027  000123  006222  CMP     #123,ITEST              ;IS IT AN S?
1640  004614  001402                  BEQ     2$
1641  004616  020027  000115  006212  CMP     #115,ITEST              ;IS IT A M?
1642  004624  001403                  BEQ     QUANT           ;GO TO QUANTITIVE TEST
1643  004626  104400  011603          PRINT,QMARK             ;TAINT EITHER
1644  004632  000750                  BR      1$              ;TRY AGAIN
1645  004634  012767  000011  006214  2$:     MOV     #11,KSTOR3              ;PRINT 9/LINE
1646  004642  012700  174336  RETA:   MOV     #20000,@ADCSR   ;CLR ALL FLAGS
1647  004646  004767  177356          JSR     PC,CLRCOR       ;CLR CLEAR CORE FROM OFFSET-MEMSIZ
1648  004652  017767  177350  006156  MOV     @SWR,STA        ;GET CH. FROM S.R
1649  004660  042767  177000  006150  BIC     #177000,STA     ;CLR UNWANTED BITS
1650  004666  052767  113000  006142  BIS     #113000,STA     ;SEQ,DMA,FINAL CH.,INC.MEM.
1651  004674  020767  000105  006120  CMP     #105,PROC       ;TEST SYNC SEL
1652  004702  001003                  BNE     .+10    ; BRANCH IF NOT EXTERNAL
1653  004704  052767  004000  006124  BIS     #4000,STA       ;EST EX SYNC
1654  004712  016767  006120  006122  MOV     STA,FINAL       ;LOAD VALUE OF REG
1655  004720  042767  001000  006110  BIC     #1000,STA       ;CLR FINAL CHANNEL
1656  004726  016767  006104  006104  MOV     STA,INITAL      ;LOAD VALUE OF INITIAL REG
1657  004734  016777  006054  174276  MOV     OFFSET,@ADWCR   ;LOAD OFFSET REGISTER
1658  004742  012777  177777  174260  MOV     #-1,@ADWCR      ;TAKE ONE CONVERTION
1659  004750  004767  001124          JSR     PC,ADCNVT       ;;DO IT
```

```
 1660  004750  017767  174246  006054          MOV     @RDSWR,STA      ;GET ADDRESS OF INCREMENTED LOC
 1661  004756  022777  000001  006046          CMP     #1,@STA         ;WAS LOCATION MODIFIED BY +1
 1662  004764  001411                          BEQ     GDVAL
 1663  004766  104000                          PRINT                   ; 'CORE LOCATION NOT INCREMENTED'
 1664  004770  012527                          MES43
 1665  004772  017767  174224  006026          MOV     @RDSWR,HOLD     ;SETUP ADDRESS TO PRINT
 1666  005000  104010                          PRTOCT                  ;VALUE OF SWAR
 1667  005002  013026                          HOLD
 1668  005004  000167  177624                  JMP     RETA            ;TRY AGAIN
 1669  005010  104016                  GDVAL:  TSTTKS
 1670  005012  005777  174174                  TST     @SWR            ;TEST BIT 15 FOR PRINT
 1671  005016  100405                          BMI     .+14
 1672  005020  017700  174176                  MOV     @RDSWR,R0       ;PRINT
 1673  005024  000005                          RESET
 1674  005026  000167  177602                  JMP     RETA            ;NO PRINT RETURN TO RESTART
 1675  005032  017767  174164  005766          MOV     @RDSWR,HOLD     ;GET CONTENTS OF SWAR
 1676  005040  104010                          PRTOCT
 1677  005042  013026                          HOLD
 1678  005044  104007                          SPACE                   ;PRINT ONE BY DEFAULT
 1679  005046  005367  005776                  DEC     KSTOR3          ;TEST FOR END OF LINE
 1680  005052  001005                          BNE     .+14
 1681  005054  012767  000011  005766          MOV     #11,KSTOR3      ;RESET
 1682  005062  104000                          PRINT
 1683  005064  011453                          CRLF
 1684  005066  000167  177542                  JMP     RETA
 1685                                          ;
 1686                                          ;TAKE INCREMENT MEMORY CONVERTIONS AND OUTPUT CONVERTION
 1687                                          ;RESULTS  SHOWING THE LOCATION INCREMENTED AND # OF INCREMENTS
 1688                                          ;
 1689
 1690  005072  004767  000230          QUANT:  JSR     PC,CLRCOR       ;CLEAR CORE FROM OFFSET TO MEMSIZE
 1691  005076  012777  020000  174114          MOV     #20000,@ADCSR   ;CLR ALL FLAGS
 1692  005104  017767  174102  005720          MOV     @SWR,STA        ;GET CH. FROM SWITCH REG.
 1693  005112  042767  177000  005712          BIC     #177000,STA             ;CLEAR UNWANTED BITS
 1694  005120  052767  113000  005704          BIS     #113000,STA     ;SEQ,DMA,FINAL CH,INC.MEM.
 1695  005126  022767  000105  005662          CMP     #105,PROC       ;TEST FOR SYNC SELECT
 1696  005134  001003                          BNE     .+10
 1697  005136  052767  004000  00[illegible]          BIS     #4000,STA       ;INSERT THE EXTERNAL
 1698  005144  016767  005662  00  64          MOV     STA,FINAL       ;LOAD FINAL ADDRESS
 1699  005152  042767  001000  0[illegible]  52          BIC     #1000,STA       ;CLR FINAL BIT
 1700  005160  016767  005646  00[illegible]46          MOV     STA,INITAL      ;LOAD VALUE OF INITIAL REG.
 1701  005166  016777  005616  174040          MOV     OFFSET,@ADDADR  ;LOAD INC MEM OFFSET REG.
 1702  005174  012777  176030  174022          MOV     #-1750,@ADWCR   ;TAKE 1000 (DEC)CONVERTIONS
 1703  005202  004767  000666                  JSR     PC,ADCNVT       ;TAKE CONVERTIONS
 1704                                          ;
 1705                                          ;SCALE FROM OFFSET TO MEMSIZ REPORTING ON CONSOLE DEVICE
 1706                                          ;EACH TIME AN INCREMENTED LOCATION IS ENCOUNTERED WITH
 1707                                          ;THE FOLLOWING (XXXXXX    ,# OF CONVERTIONS)WHERE XXXXXX=CORE MEMORY
 1708                                          ;LOCATION INCREMENTED AND # OF CONVERTIONS=# OF CONVERTIONS INCREMENTING
 1709                                          ;THE LOCATIONS
 1710                                          ;
 1711  00[illegible]206  005767  005600          TST     SOFLAG          ;HAS INITIAL MESSAGE BEEN TYPED
 1712  00[illegible]212  100410                  BMI     1$
 1713  00[illegible]214  104000  012746          PRINT,MES47             ;PRINT MEMORY OFFSET=
 1714  00[illegible]220  104010                  PRTOCT                  ;PRINT OFFSET
 1715  00[illegible]222  013010                  OFFSET
```

```
 1716  005224  104000  012661                    PRINT,MES46             ;PRINT HEADER
 1717  005230  005367  005556                    DEC     SOFLAG          ;SET FLAG
 1718  005234  016701  005550          1$:       MOV     OFFSET,R1       ;POINT TO TABLE
 1719  005240  104016                            TSTTKS
 1720  005242  005711                  2$:       TST     (R1)            ;ANY INCS?
 1721  005244  001005                            BNE     4$
 1722  005246  020167  005534          5$:       CMP     R1,MEMSIZ       ;AT TOP OF TABLE YET?
 1723  005252  001420                            BEQ     3$              ;IF YES GET OUT
 1724  005254  005721                            TST     (R1)+           ;UPDATE POINTER
 1725  005256  000771                            BR      2$              ;KEEP CYCLEING
 1726  005260  010167  005542          4$:       MOV     R1,HOLD         ;LOAD VALUE FOR PRINTING
 1727  005264  104010                            PRTOCT                  ;PRINT THE LOCATION INCREMENTED
 1728  005266  013026                            HOLD
 1729  005270  012767  000003  000110            MOV     #3,SPACEX       ;PRINT 3 SPACES
 1730  005276  104007                            SPACE                   ;PRINT THEM
 1731  005300  104016                            TSTTKS                  ;TEST TTY STATUS
 1732  005302  011102                            MOV     (R1),R2         ;GET DATA AND RESTORE R1
 1733  005304  104006                            BINDEC                  ;CONVERT DATA IN R2 AND PRINT
 1734  005306  104000  011453                    PRINT,CRLF
 1735  005312  000755                            BR      5$              ;CYCLE TILL DONE
 1736  005314  104000  011453          3$:       PRINT,CRLF              ;DONE THIS LOOP
 1737  005320  104016                            TSTTKS                  ;TEST TTY STATUS
 1738  005322  000167  177544                    JMP     QUANT           ;KEEP RUNNING
 1739                                    ;
 1740                                    ;
 1741
```

```
                                        ;SUBROUTINE TO 'CLR' CORE BEFORE USING 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 THE CONTENTS OF 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     INFLG+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  120127  000137                  CMPB    #137,R1
005456  100415                          BMI     SPCHR
005460  110124                  INPUTB: MOVB    R1,(R4)+        ;SAVE CHARACTER
005462  005267  005332                  INC     CHRCNT          ;INCREMENT THE CHARACTER COUNT.
005466  000167  000007  005324          JMP     @.+6            ;TYPE '?' IF TOO MANY CHAR.
005472  105777  173504          OUTPTA: TSTB    @TPS            ;ECHO CHARACTER
005476  100375                          BPL     OUTPTA
005500  110177  173500                  MOVB    R1,@TPB
005504  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'
       001006                       BNE     SPCHRP          ;NO, NOT '↑A'
       104000                       PRINT                   ;ECHO '↑A'
       011445                       CNTRLA                  ;RESTORE 'SP'
       022626                       POP2SP
       104021                       GETREG
       000177 005244                JMP     @AVECTR         ;YES, EXIT VIA '↑A' VECTOR ADDRESS.
       022701 000020        SPCHRP: CMP     #20,R1          ;CHAR. ='↑P'
       001006                       BNE     SPCHR1          ;NOT '↑P'
       022626                       POP2SP                  ;YES, RESTORE 'SP'
       104000                       PRINT
       011450                       CNTPLP                  ;TEST '↑P'
       104021                       GETREG
       000167 001222                JMP     DCODER          ;EXIT TO COMMAND DECODER
       122701 000003        SPCHR1: CMPB    #3,R1           ;CHAR.='↑C'
       001002                       BNE     .+6             ;NO,NOT '↑C'
       000167 173776                JMP     MONITR          ;YES, EXIT TO MONITOR
       122701 000177                CMPB    #177,R1         ;C[illegible] = 'RUBOUT'
       001011                       BNE     SPCHR2          ;IG[illegible] CHAR. & EXIT
       005767 005216                TST     CH[illegible]NT ;IS R[illegible]UT LEGAL?
       001711                       BEQ     IN[illegible]UTA ;NO, I[illegible]E IT
       005367 005210                DEC     CH[illegible]NT
       012701 000134                MOV     #13[illegible],R1 ;TYPE '\' TO INDICATE RUBOUUT
       005744                       TST     -(R4)           ;POP OFF LAST CH[illegible]ACTER
       005727                       TST     OUTPTA          ;WAIT FOR NEXT CHARACTER
       12[illegible]01 000053       CMPB    #'3,R1          ;TEST FOR '+'
       001004                       BNE     SPCHR3          ;[illegible]NCH IF NO
       01[illegible]7 000177 005134 MOV     #177,@OSIGN     ;YES, INDICATES UNIPOLAR
       0[illegible]0                BR      OUTPTA          ;WAIT [illegible]XT CHAR.
       12[illegible]01 000054       CMPB    #54,R1          ;TEST FOR ','
       001[illegible]06             BEQ     INPUTB          ;LEGAL CHAR., SAVE IT
       12[illegible]01 000015       CMPB    #15,R1          ;=TO 'CA[illegible]GE RETURN' TO TERMINATE?
       001004                       BNE     SPCHR4          ;NO, CONTINUE
       10[illegible]3               PRINT                   ;YES, TYPE 'CR-LF'
       011[illegible]3              CRLF
       10[illegible]3       EXTTY:  GETREG
       0[illegible]102              RTI
       104020               SPCHR4: PRINT                   ;OTHERWISE TYPE '?'
       011[illegible]3              GA[illegible]OK
       0[illegible]3                BR XTTYIN               ;WAIT FOR NEW ENTRY
       0[illegible]0        INBUF:  0                       ;CHARACTER STORAGE BUFFER
       [illegible]710               .=.+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  001436                          BLE     BCDEND          ;YES, EXIT
005742  005367  005052                  DEC     CHRCNT          ;DECREMENT CHARACTER COUNTER
005746  122714  000054                  CMPB    #',(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  011[illegible]                  QMARK                   ;TYPE '?'.
006062  000167  177626                  JMP     BCDBIN+2
006066  000000                  BCDTAB: 0                       ;OCTAL STORAGE TABLE
006070  000000                          0
006072  000000                          0
```

```
1879
1880                                    ;SUBROUTINE TO 'N' NUMBER OF A/D CONVERSIONS USING EITHER SEQUENTIAL
1881                                    ;OR RANDOM MODE.  ROUTINE IS ENTERED WITH 'N' IN COUNT AND THE
1882                                    ;CHANNEL AND GAIN TO BE CONVERTED IN 'INITAL' & 'FINAL' ADDRESSES.
1883
1884  006074  052777  020000  173116  ADCNVT: BIS     #20000,@ADCSR   ;CLR ALL.
1885  006102  016777  004702  173124          MOV     OFFSET,@ADBUFR  ;LOAD OFFSET REG.
1886  006110  012777  013336  173104          MOV     #MUX!IE,@ADCSR  ;LOAD STATUS WORD REGISTER
1887  006116  012777  013434  173104          MOV     #MUX!IE!1,@ADCSR ;LOAD A/D WORD REGISTER 'A'
1888  006124  017777  173074  173100          MOV     @ADBUFR,@ADBUFB ;OFFSET BUFFER 'B' VIA OF NC. OF CONVERSIONS
1889  006132  005477  173074                  NEG     @ADBUFB         ;MAKE NO. POS.
1890  006136  006377  173070                  ASL     @ADBUFB         ;OFFSET X2
1891  006142  062777  013434  173062          ADD     #ADBUFF,@ADBUFB ;FROM THE 'A' BUFFER
1892  006150  004767  002540                  JSR     PC,LDINTR       ;LOAD THE A/D INTERRUPT VECTOR.
1893  006154  006210                          CNVT1                   ;TO INTERRUPT HERE
1894  006156  052767  001000  004652          BIS     #1000,FINAL     ;SET UP FINAL CH.
1895  006164  005767  004644                  TST     INITAL          ;RUNNING SEQ. MODE?
1896  006170  100003                          BPL     .+10            ;NO, DON'T LOAD FINAL CH.
1897  006172  016777  004640  173016          MOV     FINAL,@ADCR     ;LOAD FINAL CH.
1898  006200  016777  004630  173010          MOV     INITAL,@ADCR    ;LOAD INITIAL CH. & ST. CONVERTER.
1899  006206  000001                          WAIT                    ;WAIT FOR INTERRUPT.
1900
1901                                    ;ENTERED HERE ON THE INTERRUPT
1902
1903  006210  022626                  CNVT1:  POP2SP                  ;RE-SET STACK
1904  006212  004767  002532                  JSR     PC,CLRINT       ;CLR INTR. ADDR.
1905  006216  005777  172776                  TST     @ADCSR          ;TEST ERROR BIT
1906  006222  100402                          BMI     CNVT2           ;BRANCH IF NOT SET
1907  006224  104001                          ERRINT                  ;OTHERWISE TYPE ERROR
1908  006226  011620                          MES12                   ;TEST 'ERROR BIT SET'
1909  006230  000207                  CNVT2:  RTS     PC              ;EXIT
1910
1911
1912                                    ;POWER FAIL HANDLER
1913
1914  006232  010046                  PWRFAL: MOV     R0,-(SP)
1915  006234  010146                          MOV     R1,-(SP)
1916  006236  010246                          MOV     R2,-(SP)
1917  006240  010346                          MOV     R3,-(SP)
1918  006242  010446                          MOV     R4,-(SP)
1919  006244  010546                          MOV     R5,-(SP)
1920  006246  016746  171552                  MOV     24,-(SP)
1921  006252  011267  004540                  MOV     SP,PROC
1922  006256  012767  006266  171540          MOV     #PWRUP,24
1923  006264  000000                          HALT
```

```
                                        ;POWER UP HANDLER
                                        ;
006266 012777 000340 172704 PWRUP:  MOV     #340,@PSW
006274 016706 004516        PROC,SP
006300 012667 171520        MOV     (SP)+,R4
                            MOV     (SP)+,R5
                            MOV     (SP)+,R4
                            MOV     (SP)+,R3
                            MOV     (SP)+,R2
                            MOV     (SP)+,R1
                            MOV     (SP)+,R0
                            CLR     R5
                            INC     R5
                            BNE     .-2
                            PRINT
                            MES21
006332 000167 173230        JMP     MONITR
                                        ;
                                        ;SUBROUTINE TO REQUEST A CH.(S) INPUT FROM THE TELETYPE
                                        ;
006336 104060               XCHAIN: PRINT
006340 011666                       MES14                   ;TEXT 'CH.(S)'
006342 104001                       DECOCT                  ;CONVERT TO OCTAL
                                    CLR     FINAL           ;CLR CH. STORAGE
                                    BIC     #1777,INITAL
                                    CLR     FINAL2          ;CLR 2ND CH. STORAGE
                                    BIC     #1777,INITL2
                                    MOV     BODTAB,FINAL    ;LOAD AS FINAL CH.
                                    BIS     BODTAB,INITAL   ;LOAD INITAL CH.
                                    MOV     BODTAB+2,FINAL2 ;LOAD AS 2ND FINAL CH.
                                    BIS     BODTAB+2,INITL2 ;LOAD 2ND INITIAL CH.
                                    CMP     @#PT1,AVECTR    ;ENTERED FROM REPEATIBILTY TEST?
                                    BNE     EXCHAN          ;NO, EXIT
                                    TST     BODTAB+2        ;WAS A SECOND CH. ENTERED?
                                    BEQ     EXCHAN          ;NO, EXIT
                                    CMP     FINAL2,FINAL    ;YES, IS 2ND CH. > 1ST CH.
                                    BLO     XCHAIN          ;NO, ILLEGAL ENTRY
000002                      EXCHAN: RTI
```

```
                                        ;SUBROUTINE TO INPUT A 'GAIN FROM THE TELETYPE
                                        ;
006450  104000                XGAINA: PRINT
006452  011720                        MES1B                   ;TEXT 'GAIN?'
006454  104401                        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     XGAIN3
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 TRAPPED '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  104400                      PRINT
006612  012451                      MES4O                   ;TEXT 'ILLEGAL TRAP TO'
006614  162767  000004  004232      SUB     #4,TEMP1
006622  104010                      PNTOCT
006624  013054                      TEMP1                   ;TYPE 'PC' TRAPPED TO
006626  104400                      PRINT
006630  012473                      MES41                   ;TEXT 'FROM'
006632  162767  000002  004216      SUB     #2,TEMP2
006640  104010                      PNTOCT
006642  013056                      TEMP2                   ;TYPE WHERE IT TRAPPED FROM
```

```
2018  006644  000167  172716                  JMP     MONITR          ;RETURN TO MONITOR
2019                                  ;*****************************************************************
2020                                  ;SUBROUTINE TO SAVE 'R1-R5' ON STACK
2021                                  ;*****************************************************************
2022
2023  006650  012667  004206         XSAVRG: MOV     (SP)+,SAVEPC
2024  006654  012667  004204                 MOV     (SP)+,SAVPSW
2025  006660  012667  004202                 MOV     (SP)+,SAV2PC
2026  006664  012667  004200                 MOV     (SP)+,SAV2SW
2027  006670  010146                         MOV     R1,-(SP)
2028  006672  010246                         MOV     R2,-(SP)
2029  006674  010346                         MOV     R3,-(SP)
2030  006676  010446                         MOV     R4,-(SP)
2031  006700  010546                         MOV     R5,-(SP)
2032  006702  016746  004162                 MOV     SAV2SW,-(SP)
2033  006706  016746  004154                 MOV     SAV2PC,-(SP)
2034  006712  016746  004146                 MOV     SAVPSW,-(SP)
2035  006716  016746  004140                 MOV     SAVEPC,-(SP)
2036  006722  000002                         RTI
2037
2038                                  ;*****************************************************************
2039                                  ;SUBROUTINE TO RESTORE 'R1-R5' FROM THE STACK
2040                                  ;*****************************************************************
2041
2042  006724  012667  004132         XGETRG: MOV     (SP)+,SAVEPC
2043  006730  012667  004130                 MOV     (SP)+,SAVPSW
2044  006734  012667  004126                 MOV     (SP)+,SAV2PC
2045  006740  012667  004124                 MOV     (SP)+,SAV2SW
2046  006744  012605                         MOV     (SP)+,R5
2047  006746  012604                         MOV     (SP)+,R4
2048  006750  012603                         MOV     (SP)+,R3
2049  006752  012602                         MOV     (SP)+,R2
2050  006754  012601                         MOV     (SP)+,R1
2051  006756  016746  004106                 MOV     SAV2SW,-(SP)
2052  006762  016746  004100                 MOV     SAV2PC,-(SP)
2053  006766  016746  004072                 MOV     SAVPSW,-(SP)
2054  006772  016746  004064                 MOV     SAVEPC,-(SP)
2055  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  104003                          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  013316                  AVGTAB: GPSOX1
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
```

```
2104
2105
2106                                    ;EMT DISPATCH SERVICE ROUTINE
2107                                    ;ARGUMENT OF EMT IS EXTRACTED AND USED AS OFFSET TO OBTAIN POINTER
2108                                    ;TO THE SELECTED SUBROUTINE.
2109
2110  007166  011646                    EMTSRV: MOV     (SP),-(SP)      ;GET PC FOR TO RETURN
2111  007170  162716  000002                    SUB     #2,(SP)         ;PC OF EMT
2112  007174  017616  000000                    MOV     @(SP),(SP)      ;GET EMT
2113  007200  005716                            TST     (SP)            ;IS EMT VALID?
2114  007202  001001                            BNE     EMTOK
2115  007204  000000                            HALT                    ;INVALID EMT
2116  007206  006316                    EMTOK:  ASL     (SP)            ;MULTIPLY EMT ARG BY '2'
2117  007210  042716  177001                    BIC     #177001,(SP)    ;CLEAR UNWANTED BITS
2118  007214  062716  007226                    ADD     #EMTTAB,(SP)    ;POINTER TO SUBROUTINE ADDRESS
2119  007220  017616  000000                    MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
2120  007224  000136                            JMP     @(SP)+          ;GO TO SUBROUTINE
2121
2122                                    ;EMT DISPATCH TABLE
2123
2124  007226  007306                    EMTTAB: TYPMES                  ;MESSAGE PRINT ROUTINE
2125  007230  005710                            BCDBIN                  ;DECIMAL TO BINARY CONVERSION ROUTINE
2126  007232  010246                            XRDMEM                  ;SUBROUTINE TO READ & CATEGORIZE INC. MEM. VALUES
2127  007234  000450                            XGAINA                  ;REQUEST A 'GAIN' FROM THE TTY.
2128  007236  007604                            CMPTE                   ;SUBROUTINE TO COMPUTE THE AVG
2129  007240  007726                            CATORZ                  ;SUBROUTINE TO COMPUTE 'COUNT SPREAD'
2130  007242  007404                            DECPRT                  ;SUBROUTINE TO CONVERT OCT TO DEC + PRINT
2131  007244  005356                            XSPACE                  ;SUBROUTINE TO TYPE SPACES
2132  007246  010576                            OCTPRT                  ;OCTAL PRINT ROUTINE
2133  007250  005410                            XTTYIN                  ;TELEPRINTER SERVICE ROUTINE
2134  007252  003766                            XWATGN                  ;GAIN TEST CONVERSION ROUTINE
2135  007254  003770                            XWATGN+2                ;GAIN TEST CONVERSION ROUTINE
2136  007256  007126                            XPRTAV                  ;SUBROUTINE TO PRINT OUT THE GAIN AVERAGES
2137  007260  007272                            DASH6                   ;SUBROUTINE TO TYPE OUT '6' DASHES
2138  007262  010702                            TKSFLG                  ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
2139  007264  006336                            XCHAIN                  ;SUBROUTINE TO DECODE A CHANNEL FROM TTY
2140  007266  000650                            XSAVRG                  ;SUBROUTINE TO SAVE REG'S ON THE STACK
2141  007270  006724                            XGETRG                  ;SUBROUTINE TO GET REG'S FROM THE STACK
2142
2143                                    ;SUBROUTINE TO TYPE OUT 'N' SETS OF SIX DASHES
2144
2145  007272  104000                    DASH6:  PRINT
2146  007274  012430                            MES38
2147  007276  005367  003520                    DEC     COUNT
2148  007302  001373                            BNE     DASH6
2149  007304  000002                            RTI
2150
```

```
2151
2152                                    ;MESSAGE PRINT ROUTINE, ENTERED VIA EMT DISPATCH HANDLER.
2153                                    ;ROUTINE PICKS UP CONTENTS OF THE 'PC' AND USES THIS AS
2154                                    ;THE ADDRESS OF MESSAGE TO BE TYPED.
2155
2156  007306  104020                    TYPMES: SAVREG
2157  007310  017602  000000                    MOV     @(SP),R2        ;GET THE MESSAGE ADDRESS FROM START
2158  007314  062716  000002                    ADD     #2,(SP)         ;SET UP STACK TO EXIT
2159  007320  105777  171662            TYPERA: TSTB    @TPS
2160  007324  100375                            BPL     TYPERA          ;WAIT FOR TTY DONE
2161  007326  122712  000100                    CMPB    #100,(R2)       ;TEST FOR '@'
2162  007332  001002                            BNE     TYPER1          ;BRANCH IF NO EQUAL
2163  007334  104021                            GETREG
2164  007336  000002                            RTI                     ;OTHERWISE EXIT
2165  007340  122712  000045            TYPER1: CMPB    #45,(R2)        ;TEST FOR '%'
2166  007344  001403                            BEQ     TYPECL          ;IF = TYPE 'CR-LF'
2167  007346  112277  171636            TYPER2: MOVB    (R2)+,@TPB      ;OUTPUT CHAR.
2168  007352  000762                            BR      TYPERA
2169  007354  012777  000015  171626    TYPECL: MOV     #15,@TPB        ;TYPE 'CR'
2170  007362  105777  171620                    TSTB    @TPS
2171  007366  100375                            BPL     .-4
2172  007370  012777  000012  171612            MOV     #12,@TPB
2173  007376  105722                            TSTB    (R2)+           ;INCREMENT BUFFER
2174  007400  104016                            TSTTKS
2175  007402  000746                            BR      TYPERA
2176                                    ;PRINT DECIMAL VALUE IN R2
2177
2178  007404  104020                    DECPRT: SAVREG
2179  007406  012767  177774  000152            MOV     #-4,DIGCNT
2180  007414  012767  007574  000150            MOV     #DECPNT+2,DECPNT
2181  007422  012767  000240  000140            MOV     #240,ZERO
2182  007430  012767  177777  000126    TYPT1:  MOV     #-1,DIGIT
2183  007436  005267  000122            TYPT2:  INC     DIGIT
2184  007442  167702  000124                    SUB     @DECPNT,R2
2185  007446  100373                            BPL     TYPT2
2186  007450  067702  000116                    ADD     @DECPNT,R2
2187  007454  104016                            TSTTKS
2188  007456  004767  000022                    JSR     PC,DECOUT
2189  007462  005267  000100                    INC     DIGCNT
2190  007466  001002                            BNE     TYPT3
2191  007470  104021                            GETREG
2192  007472  000002                            RTI
2193  007474  062767  000002  000070    TYPT3:  ADD     #2,DECPNT
2194  007502  000752                            BR      TYPT1
2195  007504  005767  000054            DECOUT: TST     DIGIT
2196  007510  001010                            BNE     DEC1
2197  007512  022767  177777  000046            CMP     #-1,DIGCNT
2198  007520  001404                            BEQ     DEC1
2199  007522  016767  000042  000034            MOV     ZERO,DIGIT
2200  007530  000406                            BR      DEC2
2201  007532  012767  000260  000030    DEC1:   MOV     #'0,ZERO
2202  007540  052767  000260  000016            BIS     #'0,DIGIT
2203  007546  105777  171434            DEC2:   TSTB    @TPS
2204  007552  100375                            BPL     .-4
2205  007554  016777  000004  171426            MOV     DIGIT,@TPB
2206  007562  000207                            RTS     PC
```

```
2207  007564  000000                  DIGIT:  0
2208  007566  000000                  DIGCNT: 0
2209  007570  000240                  ZERO:   240
2210  007572  007574                  DECPNT: .+2
2211  007574  001750                          1000.
2212  007576  000144                          100.
2213  007600  000012                          10.
2214  007602  000001                          1.
2215
2216                                  ;COMPUTE THE RESULTS OF '1024' CONVERSIONS AS HIGH, LOW AND AVERAGE
2217
2218  007604  012701  001777          CMPTE:  MOV     #1777,R1        ;SET UP TO COMPARE '1023' NUMBERS
2219  007610  005000                          CLR     R0              ;CLR HI ORDER DIVIDEND
2220  007612  012704  013434                  MOV     #ADBUFF,R4      ;SET UP DATA BUFFER ADDRESS
2221  007616  012403                          MOV     (R4)+,R3        ;STORE 1ST VALUE AS AVERAGE
2222  007620  010367  003252                  MOV     R3,HIGH         ;HIGH
2223  007624  010367  003250                  MOV     R3,LOW          ;& LOW
2224  007630  066703  003160                  ADD     ADSIZE,R3       ;ADD OFFSET TO AVERAGE
2225  007634  012402                  GETDAT: MOV     (R4)+,R2
2226  007636  020267  003234                  CMP     R2,HIGH         ;IS NEW NO. GREATER THAN OLD NO.
2227  007642  003402                          BLE     TSLO            ;BRANCH IF NOT GREATER
2230  007644  010267  003226                  MOV     R2,HIGH         ;OTHERWISE SAVE AS NEW HIGH
2231  007650  020267  003224          TSLO:   CMP     R2,LOW
2232  007654  003002                          BGT     TAGA
2233  007656  010267  003216                  MOV     R2,LOW          ;OTHERWISE SAVE AS NEW LOW
2234  007662  066702  003126          TAGA:   ADD     ADSIZE,R2       ;ADD OFFSET TO MAKE ALL NO. POS.
2235  007666  060203                          ADD     R2,R3           ;ADD LOW ORDER
2236  007670  005500                          ADC     R0              ;ADD CARRY TO HI ORDER
2237  007672  005301                          DEC     R1
2240  007674  001357                          BNE     GETDAT          ;1024 ADDITIONS?
2241  007676  012701  000012                  MOV     #12,R1          ;YES, DIVIDE/1024
2242  007702  006200                  AVGDAT: ASR     R0
2243  007704  006003                          ROR     R3              ;SHIFT CARRY BIT INTO LO ORDER
2244  007706  005301                          DEC     R1
2245  007710  001374                          BNE     AVGDAT          ;DONE?
2246  007712  005503                          ADC     R3              ;YES, ADD REMAINDER TO LO ORDER
2247  007714  166703  003074                  SUB     ADSIZE,R3       ;SUBTRACT OFFSET TO OBTAIN REAL AVERAGE
2250  007720  010367  003170                  MOV     R3,AVRAGE       ;SAVE AS AVERAGE
2251  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     #^O00FF,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     #AVERP5,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  005361  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'
2301 010134 016767 003004 003020 CATRS:  MOV     AVGCNT,XSPRD1
2302 010142 066767 003000 003012         ADD     PLUS1,XSPRD1
2303 010150 066767 002766 003004         ADD     MINUS1,XSPRD1   ;=TO NO. COUNTS AT SPREAD OF '1'
2304 010156 016767 003000 003000         MOV     XSPRD1,XSPRD2
2305 010164 066767 002760 002772         ADD     PLUS2,XSPRD2
2306 010172 066767 002742 002764         ADD     MINUS2,XSPRD2   ;=TO NO. COUNTS AT SPREAD OF '2'
2307 010200 016767 002760 002760         MOV     XSPRD2,XSPRD3
2308 010206 066767 002740 002752         ADD     PLUS3,XSPRD3
2309 010214 066767 002716 002744         ADD     MINUS3,XSPRD3   ;=TO NO. COUNTS AT SPREAD OF '3'
2310 010222 016767 002740 002740         MOV     XSPRD3,XSPRD4
2311 010230 066767 002720 002732         ADD     PLUS4,XSPRD4
2312 010236 066767 002672 002724         ADD     MINUS4,XSPRD4   ;=TO NO. COUNTS AT SPREAD OF '4'
2313 010244 000002                       RTI                     ;EXIT
2314
2315                                    ;SUBROUTINE TO COMPUTE THE HIGH, AVERAGE AND LOW VALUES THAT WERE STORED
2316                                    ;IN MEMORY VIA THE INCREMENT MEMORY MODE
2317
2318 010246 016700 002536 XRDMEM: MOV     OFFSET,R0       ;START AT  4K OR 8K FOR 13 BITS++++++++
2319 010252 005001               CLR     R1              ;START=TO -10V
2320 010254 005002               CLR     R2              ;SET UP AS HIORDER DIVIDEND
2321 010256 005003               CLR     R3              ;SET UP AS LO ORDER DIVIDEND
2322 010260 005067 002612        CLR     HIGH
2323 010264 005067 002610        CLR     LOW
2324 010270 005067 002516        CLR     SOFLAG          ;INITIAL HEADER INDICATOR
2325 010274 011004       ROMEM1: MOV     (R0),R4 ;MOVE THE COUNTS TO TEMP (R4)
2326 010276 005304       ROMEM2: DEC     R4              ;DECREMENT TEMP
2327 010300 100414               BMI     NXADDR          ;BRANCH IF EMPTY
2328 010302 010167 002570        MOV     R1,HIGH         ;SAVE AS HIGH
2329 010306 060103               ADD     R1,R3           ;ADD VALUES WEIGHT TO LOW ORDER
2330 010310 005502               ADC     R2              ;ADD CARRY TO HIGH ORDER
2331 010312 005767 002474        TST     SOFLAG          ;CHECK IF LOW VALUE HAS BEEN SAVED
2332 010316 001367               BNE     ROMEM2          ;OTHERWISE LOOP AGAIN
2333 010320 010167 002554        MOV     R1,LOW          ;SAVE 1ST ENTRY AS LOW VALUE
2334 010324 005367 002462        DEC     SOFLAG          ;SET  FLAG TO INDICATE THAT LOW VALUE IS FILLED
2335 010330 000762               BR      ROMEM2
2336
2337 010332 005201       NXADDR: INC     R1              ;INCREMENT VALUE
2338 010334 020067 002446        CMP     R0,MEMSIZ       ;DONE?
2339 010340 001402               BEQ     1$
2340 010342 005720               TST     (R0)+           ;UPDATE POINTER
2341 010344 000753               BR      ROMEM1
2342 010346 012704 000012 1$:    MOV     #12,R4          ;SET UP TP DIVIDE BY 1024
2343 010352 006202       AVMEM:  ASR     R2
2344 010354 006003               ROR     R3              ;SHIFT CARRY INTO LOW ORDER
2345 010356 005304               DEC     R4              ;DONE?
2346 010360 001374               BNE     AVMEM           ;BRANCH IF NO.
2347 010362 005503               ADC     R3              ;YES, ADD REMAINDER TO LOW ORDER
2348 010364 010367 002524        MOV     R3,AVRAGE       ;SAVE RESULT AS AVERAGE VALUE
2349 010370 005767 002374        TST     ADSIGN          ;IF BI POLAR SET UP FOR EXTENDING THE SIGN
2350 010374 001437               BEQ     6$              ;BITS TO COMPENSATE FOR LINEAR MEMORY
2351 010376 016704 002412        MOV     ADSIZE,R4       ;ADSIZE TO DETERMINE IF SIGN IS SET
2352 010402 000241               CLC
2353 010404 006304               ASL     R4              ;1 LEFT FOR BEGINING FF SIGN BIT
```

```
                                                BIT     R4,HIGH         ;IS SIGN BIT SET IN HIGH VALUE?
                                                BNE     1$
                                                ADD     SIGEXT,HIGH     ;NO- THEREFORE ADD SIGN EXT
                                                BR      2$
                                        1$:     BIC     R4,HIGH ;YES- THEREFORE CLEAR IT
                                        2$:     BIT     R4,LOW  ;IS SIGN BIT SET IN LOW VALUE?
                                                BNE     3$
                                                ADD     SIGEXT,LOW      ;NO-THEREFORE ADD SIGN EXT
                                                BR      4$
                                        3$:     BIC     R4,LOW  ;YES-THEREFORE CLEAR IT
                                        4$:     BIT     R4,AVRAGE       ;IS SIGN BIT SET IN AVE
                                                BNE     5$              ;IF SET CLEAR IT
                                                ADD     SIGEXT,AVRAGE   ;NO-THEREFORE ADD SIGN BIT
                                                BR      6$
                                        5$:     BIC     R4,AVRAGE       ;YES-THEREFORE CLEAR IT
                                        6$:     CLR     ORLOW
                                                CLR     ORHIGH
                                                ASL     R3              ;COMPENSATE FOR ADDRESSING IN POP11
                                                ADD     OFFSET,R3       ;ADD OFFSET TO GET ADDRESS
                                                SUB     #12,R3          ;=TO AVG -5
                                                MOV     #MINU 5,R1      ;SET UP TO SAVE COUNTS
                                        AVMEM1: CMP     R3,OFFSET
                                                BGE     SET
                                                CLR     (R1)+           ;FILL WITH 0
                                                TST     (R3)+           ;UPDATE POINTER
                                                BR      AVMEM1
                                        SET:    MOV     (R3),(R1)+      ;RETRIEVE DATA
                                                CMP     R3,MEMSIZ       ;AT END OF TAGLE YET?
                                                BEQ     2$
                                                CMP     R1,#ORHIGH      ;
                                                BEQ     1$
                                                TST     (R3)+           ;UPDATE POINTER
                                                [       SET
                                                J       CATPS
                                        1$:     C       R1,#ORHIGH      ;IF AT BELOW CORE TABLE FILL BUCKETS /0'S
                                                E       1$
                                                C       (R1)+
                                                .       2$

                                        ;       TO TYPEOUT A '6' DIGIT OCTAL NO. THE 'PC' CONTAINS
                                        ;       S OF 'W  D' TO BE TYPED

                                        OCTPRT: SAVREG
                                                V       @(SP),R0        ;THE ADDRESS OF WORD TO BE TYPED
                                                0       #2,(SP)         ;SET UP STACK TO EXIT
                                                MOV     #5,R1
                                                MOV     #376,MASK       ;MASK FOR FIRST BIT
                                                R       .+4
                                        SHIFT:  ROL     (R0)
                                                ROL     (R3)
                                                ROL     (R3)
                                                MOVB    (R3),R2
                                                BICB    M  K,R2
                                                BIS     #60,R2
                                                TSTTKS
                                                BITB    #200,@TPS
```

```
2410  010[illegible]54  100374                          BPL     .-6             ;WAIT FOR PRINTER READY
2411  010[illegible]6   110277  170326                  MOVB    R2,@TPB         ;PRINT CHAR.
2412  010[illegible]2   012767  000370  000010          MOV     #370,MASK       ;MASK FOR NEXT '5' DIGITS
2413  010[illegible]0   005301                          DEC     R1
2414  010[illegible]72  001354                          BNE     SHIFT
2415  010[illegible]74  104021                          GETREG
2416  010[illegible]76  000002                          RTI
2417  010700  000376                  MASK:   376
2418
2419                                    ;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET
2420
      010[illegible]2   105777  170274          TKSFLG: TSTB    @TKS            ;FLAG SET?
      010[illegible]4   100[illegible]01                BPL     .+4             ;NO, EXIT
      010[illegible]10  100[illegible]11                TTYIN                   ;YES,INQUIRE
2424  010712  000002                          RTI
2425
2426                                    ;SUBROUTINE TO SET UP THE A/I VECTOR ADDR TO ENABLE INTERUPTS.
2427
2428  010714  017677  000000  170314  LDINTR: MOV     @(SP),@ADINT    ;LOAD INTERRUPT SERVICE ADDRESS
2429  010[illegible]2   005[illegible]716  005[illegible]02          ADD     #2,(SP)         ;SET UP STACK TO EXIT
2430  010[illegible]0   012777  000340  170304          MOV     #340,@ADLVL     ;SET A/D INTR LEVEL @7
2431  010[illegible]4   012777  000100  170256          MOV     #100,@ADCSR     ;SET INTERRUPT ENABLE
2432  010742  005[illegible]777 170232                  CLR     @PSW            ;SET PROC. PRIORITY @0
2433  010746  000207                          RTS     PC
2434
2435                                    ;SUBROUTINE TO RESET THE A/D VECTOR ADDR TO HALT ON INTERRUPTS
2436
2436  010750  012777  000340  170222  CLRINT: MOV     #340,@PSW       ;RE-SET PROC. PRIORITY @7
2437  010[illegible]56  042777  000[illegible]0   170[illegible]34          BIC     #100,@[illegible]CSR    ;CLR A/D INTR ENABLE
2438  010[illegible]64  016777  17[illegible]0   170244          MOV     ADLVL,@ADINT
2439  010772  005[illegible]777 170[illegible]2                   CLR     @ADLVL
2440  010776  000207                          RTS     PC
```

```
                                                        ;MESSAGES
011000      000                                                 .BYTE
011001      045   042101  030506  TITLE:    .ASCII  'XADF11 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)3'
011066  041505    030455  026461
011074  000017    042101  026510
011102  000501      100

011105      045   027501  020104  MES2:     .ASCII  'XA/D LENGTH? 3'
011112  042514    043516  044124
011120  020077      100

011123      045   043442  044501  MES3:     .ASCII  'X"GAIN ACCURACY TEST". SUPPLY THE FOLLOWING VOLTAGES X'
011136  020116    041501  020103
011144  020062    054503  020040
011152  020063    021124  020056
011160  020060    020120  054514
011166  020040    041510  043040
011174  020117    047514  044527
011202  020046    053440  046117
011210  020044    041507  020123
011216  000005

011221      045   047524  051440            .ASCII  " TO SELECTED CH., TYPE 'CR' TO START TEST.3"
011227  046105    041505  042524
011235  044103    020040  020056
011243  020063    020040  020065
011251  020064    020040  020040
011257  020117    020040  001101
011265  000006    041504  002123
011273  000000

011274      045   020131  020122  MES4:     .ASCII  "XTYPE LETTER ' ' TO RUN DESIRED TEST:X"
011302  000000
011304  000000
011312  000011    000000  000000
011320  000000    000000  000000
011326  000000    041504  000003
011334  000000

011342  041447    040447  044514            .ASCII  "'C'ALIBRATION, 'R'EPEATIBILITY, 'G'AIN, R'E'COVERY,"
011350  001102    020101  047511
011356  000000    000000  000002
011364  000000    000000  000004
011372  020000    043447  020047
011400  047111    020034  020032
011406  000000    047513  041526
011414  000000    000004

011415      047   025511  041516            .ASCII  "'I'NCREMENT MEMORYX3"
011422  042522    042515  052115
```

```
011430  046440  046505  051117
011436  022531     100

011441     136  022503     100  CNTRLC: .ASCII  '↑C<200>'
011445     136  040101          CNTRLA: .ASCII  '↑A<200>'
011450  050136     100          CNTRLP: .ASCII  '↑P<200>'
011453     045     100          CRLF:   .ASCII  '<200>'
011455     045  040052          ASTRIC: .ASCII  '%*<200>'
011460  027045     100          DOT:    .ASCII  '%.<200>'
011463     077  040040          QMARK:  .ASCII  '? <200>'

011466  047111  027103  046440  MES6:   .ASCII  'INC. MEM. (Y OR N)? <200>'
011474  046515  020123  042450
011502  047440  020122  042516
011510  020077     100

011513     045  041442  046101  MES7:   .ASCII  '%"CALIBRATION TEST"<200>'
011520  041111  042122  044524
011526  047117  051122  051505
011534  021124
011540  021045  051122  047503  MES8:   .ASCII  '%"RECOVERY TEST"<200>'
011546  042136  051122  052040
011554  051505  021124     100

011561     045  044103  020056  MES9:   .ASCII  '%CH. <200>'
011565     100

011567     045  042447  054047  MES10:  .ASCII  "%'E'XT. OR 'I'NT. SYNC? <200>"
011574  027124  047440  020122
011602  044447  047047  027124
011610  051440  047131  037503
011616  040040

011620  051105  047522  020122  MES12:  .ASCII  'ERROR BIT SET!<200>'
011626  044522  020124  042523
011634  020524  040040

011640  021045  042522  042520  MES13:  .ASCII  '%"REPEATIBILITY TEST"<200>'
011646  052101  041111  046111
011654  052111  020131  042524
011662  052123  040042

011666  022445  044103  024056  MES14:  .ASCII  '%%CH.(S)? <200>'
011674  024523  020077     100

011701     103  052517  052116  MES16:  .ASCII  'COUNT SPREAD' <200>'
011706  051440  051120  040506
```

```
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  020040  032063  020040
012014  031055  020040  026440
012022  020061  020040  053101
012030  026440  020061  020040          .ASCII' +1   +2   +3   +4   +5   HI@'
012036  031053  020040  026440
012044  020063  020040  032053
012052  020040  025440  020065
012060  020040  044510     100
012065     045  047520  042527  MES21:  .ASCII  'XPOWER FAILURE @'
012072  020122  040506  046111
012100  051125  020105     100
012105     045  041445  027110  MES22:  .ASCII  'XXCH.? @'
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@'
```

```
2609  012230  053060     100
2610
2611  012233     045  040507  047111  MES32:  .ASCII  'XGAIN  5.0000  2.5000  1.2500  0.6250  '
2612  012246  020040  027065  030060
2613  012254  030060  020040  027062
2614  012262  027061  032462  030060
2615  012270  020040  027062  031066
2616  012276  030065  020040
2617  012302  027060  030463  032462          .ASCII  '0.3125  0.1563  0.0781  0.3903'
2618  012310  020040  027060  032461
2619  012316  031466  020040  027060
2620  012324  033460  030470  020040
2621  012332  027060  034463  040060
2622
2623  012340  043445  044501  020116  MES33:  .ASCII  'XGAIN  VOLTAGE AVERAGEX@'
2624  012346  053040  046117  040524
2625  012354  042507  040440  042526
2626  012362  040522  042507  040045
2627
2628  012370  030445  020040  020040  MES34:  .ASCII  'X1     @'
2629  012376  040040
2630
2631  012400  031045  020040  020040  MES35:  .ASCII  'X2     @'
2632  012406  040040
2633
2634  012410  032045  020040  020040  MES36:  .ASCII  'X4     @'
2635  012416  040040
2636
2637  012420  034045  020040  020040  MES37:  .ASCII  'X8     @'
2638  012426  040040
2639
2640  012430  026455  026455  026455  MES38:  .ASCII  '------  @'
2641  012436  020040     100
2642  012441     045  054523  041516  MES39:  .ASCII  'XSYNC? @'
2643  012446  020077     100
2644  012451     045  046111  042514  MES40:  .ASCII  ;XILLEGAL TRAP TO @;
2645  012456  040077  042114  042124
2646  012464  050101  052040  020117
2647  012472     100
2648  012473     040  051106  046517  MES41:  .ASCII  ; FROM @;
2649  012500  040040
2650  012506  044445  051516  043125  MES42:  .ASCII  'XINSUFFICIENT MEMORY@'
2651  012514  052040  042523  047105
2652  012522  042440  044515  047515
2653  012530     102
2654  012531  047503  042522  MES43:  .ASCII  'XCORE MEMORY NOT INCREMENTED=@'
2655  012536  046440  052040  051117
2656  012544  020131  047516  020124
2657  012552  047111  051103  020105
2658  012560  047105  042524  033440
2659  012566     103
2660  012567  047111  051103  MES44:  .ASCII  'XINCREMENT MEMORY TEST@'
2661  012574  046440  047105  020124
2662  012602  042515  047515  052522
2663  012610  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  020104  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
```

```
                                        .EVEN

                                        ;ADDRESS AND CONSTANTS TABLE

012770  000000  ADSIGN: 0                       ;UNIPOLAR=0, BIPOLAR=1
012772  000000          SIGEXT: 0               ;SIGN EXT .(INC MEM.)
012774  000000  INTFLG: 0
012776  001000  STACK:  1000                    ;INITIAL SP. ADDRESS
013000  001540  AVECTR: INITB                   ;'↑A' VECTOR ADDRESS
013002  001722  PVECTR: INIT3                   ;'↑P' VECTOR ADDRESS
013004  000000  INCFLG: 0                       ;SOFTWARE FLAG;0=NO INC. MEM.
013006  000000  MEMSIZ: 0                       ;CALCULATED MEM SIZE TO SUPPORT INC. MEM
013010  000000  OFFSET:0                        ;CALCULATED MEMORY OFFSET.↔↔↔↔↔↔
013012  000000  SOFLAG: 0                       ;SOFTWARE 'FLAG'
013014  000000  ADSIZE: 0                       ;OCTAL STORAGE OF A/D LENGTH
013016  000000  PROC:   0                       ;TEMP STORAGE FOR 'PSW'
013020  000000  CHCNT:  0                       ;TEMP STORAGE
013022  000000  COUNT:  0                       ;TEMP STORAGE
013024  000000  ICOUNT: 0                       ;TEMP STORAGE
013026  000000  HOLD:   0
013030  000000  ITEST:  0
013032  000000  STA:    0
013034  000000  INITAL: 0                       ;PERMANENT STORAGE
013036  000000  FINAL:  0                       ;PERMANENT STORAGE
013040  000000  INITL2: 0                       ;PERMANENT STORAGE
013042  000000  FINAL2: 0                       ;PERMANENT STORAGE
013044  000000  KSTOR1: 0                       ;PERMANENT STORAGE
013046  000000  KSTOR2: 0                       ;PERMANENT STORAGE
013050  000000  KSTOR3: 0                       ;PERMANENT STORAGE
013052  000000  KSTOR4: 0                       ;PERMANENT STORAGE
013054  000000  TEMP1:  0                       ;TEMPORARY STORAGE
013056  000000  TEMP2:  0                       ;TEMPORARY STORAGE
013060  000000  TEMP3:  0                       ;TEMPORARY STORAGE
013062  000000  SAVEPC:
013064  000000  SAVPSW:
013066  000000  SAV2PC:
013070  000000  SAV2SW:
013072  000000  RETSW:
013074  000000  RESART:
013076  000000  HIGH:
013100  000000  LOW:
013102  000000  AVERP5:
013104  000000  AVERP4:
013106  000000  AVERP3:
013110  000000  AVERP2:
013112  000000  AVERP1:
013114  000000  AVERPE:
013116  000000  AVERM1:
013120  000000  AVERM2:
013122  000000  AVERM3:
013124  000000  AVERM4:
013126  000000  AVERM5:
013130  000000  ORLOW:
013132  000000  MINUS5:
013134  000000  MINUS4:
013136  000000  MINUS3:
```

```
 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                    POS062: 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                    GP25X1: 0
 2766  013220  000000                    GP12X1: 0
 2767  013222  000000                    GP62X1: 0
 2768  013224  000000                    GP31X1: 0
 2769  013226  000000                    GP25X2: 0
 2770  013230  000000                    GP12X2: 0
 2771  013232  000000                    GP62X2: 0
 2772  013234  000000                    GP31X2: 0
 2773  013236  000000                    GP15X2: 0
 2774  013240  000000                    GP12X4: 0
 2775  013242  000000                    GP62X4: 0
 2776  013244  000000                    GP31X4: 0
 2777  013246  000000                    GP15X4: 0
 2778  013250  000000                    GP07X4: 0
 2779  013252  000000                    GP62X8: 0
 2780  013254  000000                    GP31X8: 0
 2781  013256  000000                    GP15X8: 0
 2782  013260  000000                    GP07X8: 0
 2783  013262  000000                    GP03X8: 0
 2784  013264  000000                    GM50X1: 0
 2785  013266  000000                    GM25X1: 0
 2786  013270  000000                    GM12X1: 0
 2787  013272  000000                    GM62X1: 0
 2788  013274  000000                    GM31X1: 0
 2789  013276  000000                    GM25X2: 0
 2790  013300  000000                    GM12X2: 0
 2791  013302  000000                    GM62X2: 0
 2792  013304  000000                    GM31X2: 0
 2793  013306  000000                    GM15X2: 0
 2794  013310  000000                    GM12X4: 0
 2795  013312  000000                    GM62X4: 0
 2796  013314  000000                    GM31X4: 0
 2797  013316  000000                    GM15X4: 0
```

```
 2798  013320  000000                    GM15X4: 0
 2799  013322  000000                    GM07X4: 0
 2800  013324  000000                    GM62X8: 0
 2801  013326  000000                    GM31X8: 0
 2802  013330  000000                    GM15X8: 0
 2803  013332  000000                    GM07X8: 0
 2804  013334  000000                    GM03X8: 0
 2805                                    ;HERE STARTS A '30' WORD STATUS WORD BUFFER
 2806
 2807  013336  000000                    RAWBUF: 0
 2808          013434                            .=.+60.
 2809
 2810                                    ;HERE STARTS THE '512' WORD A/D DATA BUFFER.
 2811  013434  000000                    ADBUFF: 0
 2812
 2813          001242                            .END    INIT
```

```
RDRDR   001234            871#  1071#    57#  1701#  1895#
RDBUFF  013434           1023   1031    1501   1897   1891   2220   2275   2811#
RDCNVT  006074           1020   1037    1510   1593   1659   1703   1884#
RDCR    001216            864#  1897#  1898#
RDCSR   001220            865#  1646#  1691#  1884#  1905   2430#  2437#
RDDBR   001226            868#
RDINT   001236            872#  2427#  2429#
RDLVL   001240            873#  2429#  2438   2439#
RDSIGN  012770            883#   913   1181   1207   1243   1291   1345   1393   1429   1453   1489   1818#  2349
                         2690#
RDSIZE  013014            932#   935   2224   2232   2243   2250   2254   2260   2264   2351   2700#
RDSUB   001262            866#  1660   1665   1672   1675   1886#
RDUCR   001254            867#  1018#  1085#  1509#  1592#  1658#  1702#  1888
RDUBA   001230            869#  1687#
RDUBE   001232            870#  1879#  1889#  1890#  1891#
RSTRIC  011456           2061   2508#
RVECTR  013000            958#   990#  1055#  1164#  1570#  1622#  1797   1954   2694#
RVERR1  013112           1518   2731#
RVERR2  013110           2730#
RVERR3  013106           2729#
RVERR4  013104           2728#
RVERR5  013102           2281   2284   2727#
RVERP1  013116           1516   2251   2733#
RVERP2  013120           2734#
RVERP3  013122           2735#
RVERP4  013124           2736#
RVERP5  013126           2279   2737#
RVGCNT  013144           1134   2301   2744#
RVGDAT  007702           2238#  2241
RVGTAB  007136           1023#  1466#  1601#  2097#  2098#
RVMEM   010662           2343#  2346
RVMEM1  010622           2375#  2379
RVRADX  013114           1125   1512   1514   1556   2244#  2249   2259   2261   2348#  2364   2366#  2368#  2732#
                                       2125
RCDBIN  005710           1836#  1874
RCDBN1  005730           1841#  1865
RCDBN2  005734           1843#  1861
RCDEN0  006016           1844   1847   1862#
RCDERR  006054           1849   1851   1867   1869   1872#
RCDTAB  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  001764             996   999#
CALB2A  002120            1016   1018#  1038
CALB2B  002174            1022   1031#
CATORI= 104005             841#  1093   1513
CATOR2  007726            2129   2248#
CATRI   010022            2271#  2273
CATRI2  010042            2276#  2293   2295   2297
CATRI3  010072            2285#  2289
CATRI4  010110            2286   2291#
CATRI5  010134            2277   2301#  2387
```

```
CHKCNT  002[illegible]  1102  1108#
CH[illegible]NT  013[illegible]  1771#  17[illegible]#  1783  1810  1812#  1843  1845#  1864  2702#
CLRCOR  0[illegible]6  1084  1647  1690  1744#
CLRCR1  0[illegible]340  1747#  1749
CL[illegible]INT  010[illegible]30  954  1904  2436#
C[illegible]TE  0[illegible]4  2128  2218#
C[illegible]UTE=  104[illegible]4  840#  10[illegible]  1511
CN[illegible]LA  011445  1794  [illegible]#
C[illegible]LC  011441  976  [illegible]#
CNTRLP  011450  1802  25[illegible]#
[illegible]  [illegible]  1893  1903#
[illegible]  [illegible]  1906  1909#
[illegible]  [illegible]  914  916#
[illegible]  [illegible]  1476#  14[illegible]1#  1486#  1581#  2147#  2703#
C[illegible]F  011[illegible]3  1003  10[illegible]9  1111  1117  1137  1496  1525  1635  1683  1734  1736  1825  2506#
C[illegible]6  (7272  2137  2145#  2148
DECIMER  007[illegible]30  1804  20[illegible]0#  2072  2076  2078  2086  2091
DECOCT=  104[illegible]31  837#  [illegible]  1076  1945  1966  2089
DECOUT  007504  2188  21[illegible]5#
DECPNT  007572  21[illegible]#  2184  2186  2193#  2210#
DECPRT  007404  2130  2178#
DEC1  007532  21[illegible]6  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  007206  2114  2116#
EMTSRV  007166  829  2110#
EMTTRP  007226  2118  2124#
ERTRAP  006576  821  20[illegible]5#
EXCHAN  006446  1955  1957  1960#
EXCOMP  005354  1745  1751#
EXGAIN  006510  1972  1975#
EXTTY  00[illegible]56  1826#
FILE1  007746  2252#  2256
FILE2  010002  22[illegible]#  2[illegible]6
FINAL  013036  10[illegible]5#  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  8[illegible]9#  1073  1580  2077
G[illegible]EXT  004134  1515  1517  1519  1522#
GOVRAL  005010  1662  1664#
G[illegible]1  004150  1521  1527#
G[illegible]2  004200  1530  1534#
G[illegible]3  004214  1535  15[illegible]#
G[illegible]4  00[illegible]2  15[illegible]0  15[illegible]#
G[illegible]5  00[illegible]0  15[illegible]5  15[illegible]#
G[illegible]6  00[illegible]4  15[illegible]3  1543  1548  1551#
GETCHR=  10[illegible]17  [illegible]18  10[illegible]2  1169  1579  2075
GETDAT  007634  2[illegible]58  2[illegible]6
GET[illegible]G=  104[illegible]21  [illegible]38  17[illegible]6  1803  1826  18[illegible]0  2163  2191  2415
GLOOP  004[illegible]46  15[illegible]8  1[illegible]  1538
G[illegible]3X8  013334  14[illegible]2  [illegible]
GMU7X4  013322  14[illegible]  2[illegible]
```

```
CM07X8  013332            1444    2803#
CM12X1  013272            1252    2787#
CM12X2  013302            1258    2791#
CM12X4  013312            1264    2795#
CM15X2  013322            1402    2794#
CM15X4  013332            1408    2798#
CM15X8  013342            1414    2802#
CM2X1   013270            1216    2786#
CM2X2   013300            1222    2790#
CM31X1  013270            1354    2789#
CM31X2  013300            1360    2793#
CM31X4  013310            1366    2797#
CM31X8  013320            1372    2801#
CM50X1  013266            1191    2785#
CM6X1   013274            1300    2788#
CM6X2   013304            1306    2792#
CM6X4   013314            1312    2796#
CM6X8   013324            1318    2800#
CM8     013264            1452    2784#
CM7X4   013252            1422    2779#
CM7X8   013262            1428    2783#
CM12X1  013226            1230    2767#
CM12X2  013236            1236    2771#
CM12X4  013246            1242    2775#
CM15X2  013256            1250    2774#
CM15X4  013266            1256    2778#
CM15X8  013276            1262    2782#
CM2X1   013220            1200    2766#
CM2X2   013230            1206    2770#
CM31X1  013240            1332    2769#
CM31X2  013250            1338    2773#
CM31X4  013260            1344    2777#
CM31X8  013270            1350    2781#
CM50X1  013216            1180    1466    2097    2765#
CM6X1   013224            1272    2769#
CM6X2   013234            1278    2772#
CM6X4   013244            1284    2776#
CM6X8   013254            1290    2780#
CT      013176            1163    1167#
CT0     013200            1165    1169#   1491    1494
CT1     013202            1171    1175#
CT10    013246            1471#   1497
CT11    013250            1480    1483#
CT2     013204            1182    1186#
CT3     013206            1194    1198#
CT4     013210            1206    1210#
CT5     013212            1218    1222#
CT6     013214            1230    1234#
CT7     013216            1242    1246#
CT8     013220            1454    1458#
CT9     013244            1466    1470#
        =  013176         1178    1189    1198    1214    1228    1250    1270    1298    1324    1352
        =  013176         1204    1220    1234    1256    1276    1304    1330    1358    1378    1400    1539    1984
        =  013176         1240    1262    1292    1310    1336    1364    1384    1406    1420    1436    1544    1989
        =  013176         1288    1316    1342    1370    1390    1412    1426    1442    1450    1460    1994
XIOM    013076            1128    2222#   2226    2228#   2322#   2328#   2354    2356#   2358#   2725#
```

```
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    928#
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#   979   2694
INITL2  013040          1575*  1576*  1583   1949*  1953*  1973   2710#
INIT1   001240           938#   900    941
INIT2   001644           975    978#   983
INIT3   001722           959    978#  2695
INPUTA  005426          1773#  1774   1788   1811
INPUTB  005460          1781#  1821
INTFLG  012774          2692#
INTVC=  000004           552#
ITEST   013030          1636*  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          1093*  1118   1145*  1146   2715#
LDINTR  010714          1642   2427#
LOOP    001346           909#   912
LOOPS   001340           903#   906
LDSIZE  001342           890    895   901#
LP24    013100          1122   2223*  2229   2231*  2323*  2333*  2359   2361*  2363*  2726#
MASK    010700          2400*  2406   2412*  2417#
MAX12   013016           930*  1722   1748   1750*  2338   2381   2697#
MAX24   013074          1060*  1112   1129   1133*  1171*  1529   1531*  2724#
MAXO    011567           965   1631   2531#
MIN20   011620          1908   2538#
MIN3    011640          1057   2543#
MIN4    011646          1944   2547#
MIN5    011701          1075   2553   2551#
MIN6    011730          1965   2558
MIN7    011736          1115   2563
MIN8    011744           885   2568
MIN9    011764          1132   2574#
MIN10   012065          1938   2576#
MIN11   012105          2500   2581#
MIN12   012115          1176   2583#
MIN13   012124          1187   1212   1248   1296   1350   1398   1434   1458   2586#
MIN14   012141          1196   2593
MIN15   012150          1206   2598
MIN16   012157          1213   2603
MIN17   012167          1223   2608
MIN18   012200          1376   2613#
MIN19   011123          1166   2618#
MIN20   012211          1418   2623#
MIN21   012222          1448   2628#
```

```
POS125  013176          1229    1277    1337    1391    2757#
POS250  013174          1199    1235    1283    1343    2756#
POS312  013202          1325    1379    1421    1451    27[illegible]3#
POS[illegible]0  013172          901    1179    12[illegible]5    1241    1[illegible]3    2755#
POS[illegible]5  013[illegible]00          1271    1331    1[illegible]3    14[illegible]7    2[illegible]8#
PRINT = 104000          836#    [illegible]1    8[illegible]4    8[illegible]3    940    943    955    960    978    992    994    997    1002
                        1028    10[illegible]6    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    1[illegible]31    1824    1828    1872    1907    1937    1943    1964    2008    2013    2060    2070
                        20[illegible]3    2[illegible]37    2145
PROC    013016          1[illegible]3#    1015    1633#    1651    1695    1920#    1926    20[illegible]5#    2701#
PRTADR  004256          1504#    1514    1516    1518    1552#
PRTAVG= 104014          848#    1024    1473    1478    1483    14[illegible]9    1602
PRTOCT= 104010          844#    1121    1124    1127    1551    15[illegible]5    1666    1676    1714    1727    2011    2016    2096
PSW     001200          857#    880#    1618#    1925#    2431#    2436#
PU[illegible]90= 01[illegible]046          555#
PU[illegible]15= 0[illegible]046          5[illegible]3#
PU[illegible]2= 0[illegible]046          5[illegible]7#
PV[illegible]IR  013002          [illegible]9#    991#    1061#    1168#    1576#    1621#    2081    2695#
PW[illegible]FAIL  00[illegible]232          823    1913#
PW[illegible]UP  00[illegible]266          1921    1925#
QU[illegible]X  011463          899    940    979    1643    1629    1873    2071    2512#
QU[illegible]T  005072          1642    1690#    1738
RE[illegible]UF  013336          1[illegible]5    1[illegible]    2607#
RC[illegible]M = 104002          [illegible]#    1[illegible]
RD[illegible]MI  010274          2[illegible]#    2341
RD[illegible]MI2  010276          2[illegible]#    2[illegible]2    2335
RECVRY  004276          974    1570#
RECVY1  004310          1570    1573#
RECVY2  004354          1576    1[illegible]1#
RECVY3  004372          1594#    1[illegible]
RECVY4  004406          1[illegible]8#    15[illegible]
RECVY5  004416          1591#    15[illegible]    1603
REPTST  002226          [illegible]6    1055#
REPT1  002240          1055    1058#    1954
REPT1A  002330          1063    1068    1072#
REPT2  002350          1061    1078#
REPT2A  002364          1080#    1146
REPT3  002414          1084#    1147
REPT3A  002446          1089    1092#
REPT3B  002462          1091    1094#
REPT4  002524          1095    1107    1110#
REPT5  002562          1113    1116#
REPT6  002620          1130    1133#
REPT6A  00[illegible]646          1139#    1142
REPT7  00[illegible]660          1097    1109    1135    1143#
RETA    00[illegible]634          1621    16[illegible]6#    16[illegible]8    1674    1684
RETSAV  013072          1499#    15[illegible]2    1[illegible]7#    2723#
R0      =000000          535#    9[illegible]1#    9[illegible]3#    9[illegible]5    9[illegible]9#    911    9[illegible]4#    935    9[illegible]8#    1031#    10[illegible]#    1101    1103#
                        1672#    1[illegible]3#    1[illegible]0    1913    1933#    2219#    2234#    2238#    2275#    2278    2318[illegible]    2325    2338
                        2340    2[illegible]7#    2[illegible]#    2403#    2404#    2405
R1      =000001          5[illegible]#    [illegible]7#    [illegible]#    915#    916#    921    923#    9[illegible]5#    9[illegible]6    930    933#    9[illegible]7#    942
                        1[illegible]#    1[illegible]4#    1[illegible]#    1104    1105    1108    15[illegible]2#    15[illegible]4#    15[illegible]#    1718#    1720    1722    17[illegible]4
                        1726    1732    1746#    1747#    1748    1775#    1776#    1777    1779    1781    1787    179[illegible]    17[illegible]8
```

```
                    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      =X000002      537*    892*    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    2008    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      =X000003      538*    891*    896*    902*    903     904*    907*    908*    909     910*    1839*   1863*   1916
                    1930*   2029    2048*   2221*   2222    2223    2224*   2233*   2234*   2242*   2243*   2244    2251*
                    2253*   2254*   2261*   2263*   2264*   2270*   2271*   2272    2274*   2275*   2321*   2329*   2344*
                    2347*   2348    2371*   2372*   2373*   2375    2378    2380    2381    2385
R4      =X000004      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      =X000005      540*    1918    1928*   1934*   1935*   2031    2046*
SAVEPC  013062        2023*   2035    2042*   2054    2719*
SAVPSW  013064        2024*   2034    2043*   2053    2720*
SAVREG= 104020        852*    1769    1835    2155    2178    2396
SAV2PC  013066        2025*   2033    2044*   2052    2721*
SAV2SW  013070        2026*   2032    2045*   2051    2722*
SET     010546        2376    2380*   2386
SWTFT   010654        2402*   2414
SIGEXT  012772        942*    2356    2361    2376    2691*
SIXOSW= 104015        849*    1477    1482    1487
SIZE    001316        892*    897
SOFLAG  013012        1058*   1069*   1079    1080*   1711    1717*   2324*   2331    2334*   2699*
SP      =X000006      541*    879*    903*    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
                    2193*   2307    2308*   2427    2483*
SPACE = 104007        883*    1110    1123    1126    1553    1554    1600    1678    1730    2100
SPACEX  005406        1729*   1741*   1763*   1765*   2099*
SPCHR   005512        1778    1790    1791*
SPCHRP  013034        1792    1795
SPCHR1  005516        1793    1801
SPCHR2  013040        1809    1814
SPCHR3  005526        1817    1820*
SPCHR4  013044        1822
SPCHR5  005662        1784    1823    1828*
STA     013032        1648*   1649*   1650*   1653*   1654    1655*   1656    1660*   1661    1692*   1693*   1694*   1697*
                    1698    1699*   1700    2707*
STACK   012776        879     953     1619    2693*
SWR     001212        882*    1015    1007    1010    1014    1021    1037    1094    1096    1520    1527    1594    1648
                    1670    1682
SWR0    001214        883*
S[illegible]D   =   000001  891*
S[illegible]1   =   000002  892*
S[illegible]2   =   000004  893*
S[illegible]3   =   000010  894*
S[illegible]4   =   000020  895*
SWR5    =   000040  896*
```

```
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            647#  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            851#  1760#  1787#  2167#  2169#  2172#  2205#  2411#
TPS     001206            850#  1758   1785   2159   2170   2203   2409
TSLO    000000           2227   2229#
TSTCNT  000006           1101#  1106
TSTCT4  000002           1096#
TSTTKS= 104016            870#  1019  1036  1086  1508  1591  1669  1719  1731  1737  2174  2187  2408
TTYIN = 104011            845#   962   999  1066  1498  1632  1637  1837  2062  2084  2423
TTYIN1  007744           2166   2169#
TTYIN2  007746           2159#  2160   2168  2175
TTYIN3  007750           2162   2165#
TTYIN4  007752           2167#
TTYIN5  007754           2124   2156#
TYPT1   007760           2182#  2194
TYPT2   007762           2183#  2195
TYPT3   007764           2190   2193#
WAITGN= 104012            870#  1177  1188  1197  1213  1227  1249  1269  1297  1323  1351  1377  1399
                         1419  1435  1449  1459
XCHKIN  005736           1943#  1949  2139
XCLOSE  005742           1918#  1923  2127
XCLOSI  005744           1889   1894  1977#
XFILSI  005746           1859   1868#
XGETIN  005750           1903   1908#
XGETSI  005752           1873   1878#
XGETTR  005754           2023#  2141
XPRTAV  007126           2045#  2136
XPTA1   007134           2096#  2102
XSCMEM  010246           2123   2318#
XSAVMG  006650           2023#  2145
XSPACE  006336           1756#  1762   2131
XSPRIN  013126           1093   1141   2301#  2302#  2303#  2304   2751#
XSPRNT  013132           2304#  2305#  2306#  2307   2752#
XSPRTB  013134           2307#  2308#  2309#  2310   2753#
XSPRTN  013136           1105   2310#  2311#  2312#  2754#
XTYIN   006412            856#  1769#  1830   2133
XWAITC  006452           1450#  2134   2135
ZERO    007750           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# 836# 922 964 967 970 973 976 1483 1523 1602
1671 1740 1696 1759 1806 1832# 1896 1936 2030 2171 2204 2210 2401
2410 2422 2808#

M06

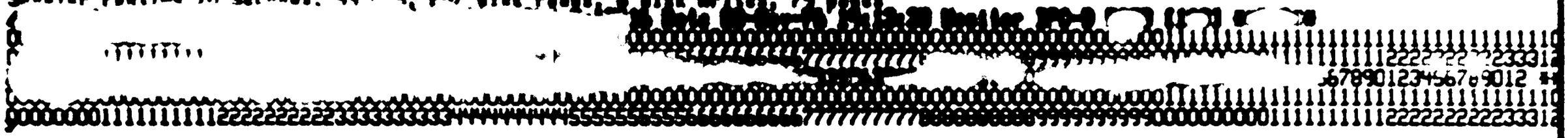

```
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$40CA      1#
.1170       1#
```

```
        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      869  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  1[illegible]2
        1864  1895  1905  1956  1970  1971  2073  2079  2113  2195  2331  2340  2349  2378  2[illegible]5
TSTB    1758  1773  1785  2159  2170  2173  2203  2421
WAIT    1899
.A S     508
.ASCII  2443  2451  2456  2460  2470  2479  2[illegible]  2495  2500  2502  2[illegible]  2[illegible]  2508  2510  2512
        2515  2[illegible]  2[illegible]  2[illegible]  2531  2538  2542  2547  2551  2555  2[illegible]  2[illegible]  2[illegible]  2576  2[illegible]
        2583  2[illegible]  2[illegible]  2[illegible]  2596  2599  2602  2605  2608  2611  2618  2624  2629  2632  2635
        2638  2641  2643  2645  2649  2651  2655  2661  2665  2672  2681
.BYTE   2442
.E[illegible]3L    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)
```

Spooler runtime 10 Seconds. [illegible] disk pages, [illegible] disk writes, [illegible] pages



```
  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
```