ADF11

LOGIC DIAGNOSTIC

MD-11-DZADG-A

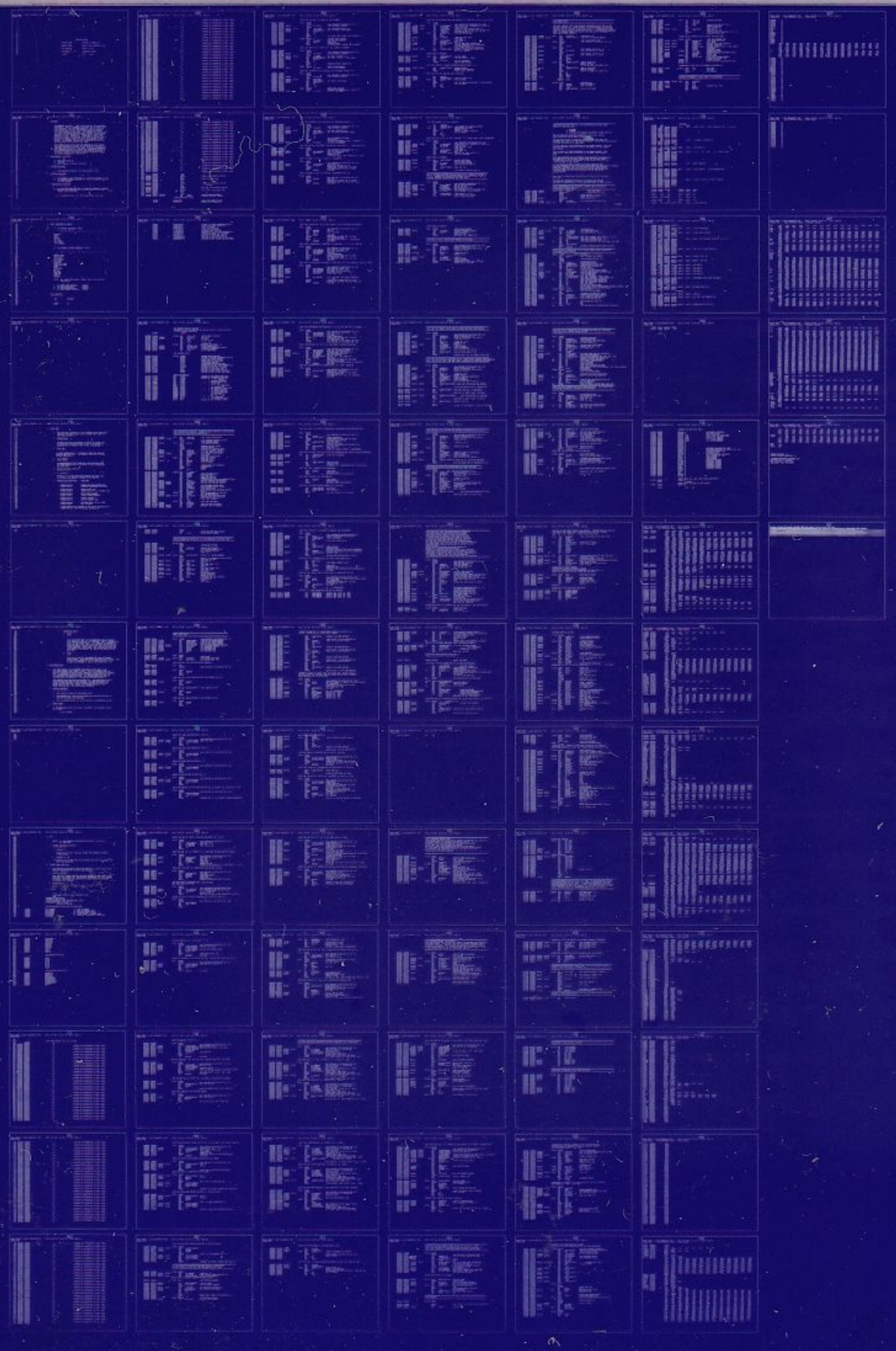
EP-DZADG-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A



```
                        .REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZADG-A-D
PRODUCT NAME:	ADF11 LOGIC DIAGNOSTIC TEST
DATE CREATED:	MARCH 4, 1974
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	EARL L. BOUSE

1. ABSTRACT

THIS PROGRAM IS PART I, THE 'LOGIC' SECTION, OF A TWO PART DIAGNOSTIC. PART II (MAINDEC 11-DZADH-A) IS THE ANALOG TEST. THIS DIAGNOSTIC TEST AND EXERCISERS THE 'ADF11' LOGIC 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 13 BIT UNIPOLAR OR BIPOLAR INPUT. EXAMPLE: 10(CR)# WOULD INDICATE A 10 BIT UNIPOLAR A/D; WHERE TYPING 10+(CR) WOULD INDICATE A 10 BIT BIPOLAR A/D. A SENTANCE IS THEN TYPED GIVING THE LETTER DISIGNALTORS TO BE TYPED TO RUN ANY ONE OF THE FOUR (4) SEPERATE TEST OF WHICH THIS PROGRAM IS COMPRISED. THE PROGRAM THEN WAITS IN A KEYBOARD MONITOR MODE AND WAITS FOR A TEST LETTER TO BE TYPED.

THE PROGRAM IS SET UP TO GIVE THE OPERATOR AS MUCH CONTROL OVER THE PROGRAM AS POSSIBLE VIA THE TELETYPE. TYPING A ↑C (OBTAINED VIA TYPING THE 'CNTR' AND 'C' KEYS SIMULTANEOUSLY) WHILE RUNNING ANY TEST WILL ENABLE THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR AND AWAITS A NEW LETTER DESIGNATOR TO TO 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 APPLICBLE CONSOLE SWITCH SETTINGS

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

6. ADF11 ADDRESSES & FORMATS

A. A/D CONTROL REGISTER (164006)

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

SEQ/RAN
GAIN
DMA/PC
EXT/INT
INC. MEM
FINAL/INIT.
MUX. ADDRESS

B. A/D CONTROL & STATUS REGISTER (164010)

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

ERROR & NXM
CNV IN PROG.
INIT A/D
DATA OVRFLO
CLR FLAGS
A/D WAIT
FINAL CH. FLAG
OVER FLOW CHEK
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
LOGIC	8.
DATA	9.

```
                                        T'NNN           10.
                                                                -2-
```

B. LOGIC TEST

A. THE LOGIC TEST CONSISTS OF '106' SUBTESTS WHICH CHECK AND EXERCISE THE 'ADF11' EACH TEST IS LOOPED '50' TIMES TO TEST LOGIC RELIBILITY.

B. RESTRICTIONS

EXTERNAL SYNC MUST BE JUMPERED TO GROUND TO ENSURE THAT NO EXTRANEOUS CONVERSIONS TAKE PLACE. A PROVISION IS PROVIDED IN THE 'ANALOG' TEST TO TEST THIS FEATURE.

C. TEST TIME

IT TAKES APPROXIMATELY '2' MIMUTES TO MAKE ONE COMPLETE PASS OF THE LOGIC TEST. THE MESSAGE 'LOGIC OK' IS TYPED ON PASS COMPLETION.

D. LOGIC ERRORS

ON ENCOUNTERING A LOGIC ERROR (ALL DATA SWITCHES DOWN) THE ERROR ADDRESS AND THE CONTENTS OF THE A/D REGISTERS ARE TYPED OUT.

E. CONTROL SWITCHES (TELETYPE)

1. ↑C (CONTROL C)

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

G. CONSOLE SWITCH SETTINGS FUNCTIONS

CONSOLE SW11=0	NORMAL RUN [2048 PASSES/TEST]
CONSOLE SW11=1	SURPRESS SUBPROGRAM ITERATIONS
CONSOLE SW12=0	NORMAL ERROR LOOP
CONSOLE SW12=1	ISSUE R-E-S-E-T (*) IN ERROR LOOP
COLSOLE SW13=0	PRINT ERROR MESSAGE
CONSOLE SW13=1	INHIBIT ERROR MESSAGES
CONSOLE SW14=0	INHIBIT LOOPING
CONSOLE SW14=1	LOOP ON CURRENT TEST
CONSOLE SW15=0	CONTINUE AFTER TYPING ERROR
CONSOLE SW15=1	HALT ON ERROR

* SOME SUBTESTS WILL REQUIRE A "R-E-S-E-T" INSTRUCTION TO ENABLE THE A/D TO BE INITIALIZED FOR SCOPING.

```
195
196
```

-3-

H. SUBROUTINE CALLS

1. SCOPE

THIS IS AN "EMT TRAP" TO THE SUBROUTINE CALLED "SCOPEC". THIS SUBROUTINE HAS TWO (2) FUNCTIONS. ONE, IT RECORDS THE "P.C." OF THE "SCOPE" CALL. THIS ENABLES THE PROGRAM TO ENTER THE NEXT LOGIC TEST OR TO LOOP BACK TO THE PREVIOUS "SCOPE" ADDRESS. THUS A SCOPE LOOP WILL CONTAIN ALL THE CODE BETWEEN TWO SCOPE CALLS. TWO, CHECKS THE ITERATIONS TO BE MADE ON THE CURRENT TEST BEFORE CONTINUING TO THE NEXT TEST.

2. ERROR

THIS IS A "T-R-A-P" TO A SUBROUTINE CALLED "LOGERR" (LOGIC ERROR). THIS SUBROUTINE ALSO HAS TWO FUNCTIONS. ONE, TO PRINT THE ADDRESS OF T-H-A-T ERROR DETECTED. TWO, TO HALT ON THE ERROR IF SWITCH "15" IS SET TO A "1".

9. DATA UPDATE TEST

A. THE "DATA UPDATE" IS AN OPERATOR INTERVENTION TEST USED TO THE A/D "DATA BUFFERS" AND "UPDATE" LOGIC. THE TEST REQUESTS FOUR (4) SPECIFIC VOLTAGES (2 POSITIVE AND 2 NEGATIVE) TO BE SUPPLIED TO CHANNEL "0". A CONVERSION IS TAKEN AT EACH OF THESE VOLTAGES AND THEN THE FOUR BUFFERED DATA WORDS ARE READ. THE PROGRAM CHECKS THE SIGN OF THE READ DATA TO DETERMINE IF THE DATA WAS BUFFERED CORRECTLY.

WHEN RUNNING THIS TEST WITH A UNI-POLAR A/D. THE USER SHOULD SUPPLY +1.00 VOLT FOR THE 2ND AND 3RD VOLTAGES. AN ERROR PRINTOUT WILL OCCUR ON THESE READINGS. THESE READINGS MAY THEN BE VERFIED FOR BEING CLOSE TO THE 1 VOLT VALUE. IF THE READINGS ARE A 5 VOLT VALUE, THE DATA ISN'T GETTING BUFFERED CORRECTLY.

B. STARTING SEQUENCE

1. TYPE "D<CR>" TO RUN THE "DATA UPDATE TEST".

2. TYPE PROGRAM WILL TYPE THE TEST HEADER AND THEN ASK FOR FOUR (4) SPECIFIC VOLTAGES TO BE SUPPLIED.

3. TYPE A CARRIAGE RETURN <CR> AFTER SUPPLING THE REQUESTED VOLTAGE.

C. ERROR FORMAT

IF THE READ DATA IS NOT THE POLARITY EXPECTED, THE FOLLOWING TYPEOUT WILL OCCUR:

```
        1ST RD  NNNNNN
```

```
                                                A       B

                                                        -4-
```

(9. CONT.)

WHERE: A = THE NUMBER OF THE READ WHICH FAILED (1,2,3 OR 4)
B = READ DATA

D. CONTROL SWITCHES [TELETYPE]

1. [CONTROL C]

TYPING A <↑C> AT ANY TIME WILL ENABLE THE PROGRAM TO RETURN TO THE MONITOR.

2. [CONTROL A] <↑A>

TYPING A <↑A> AT ANY TIME WHILE RUNNING THIS TEST WILL ENABLE THE TEST TO BE RESTARTED.

10. T'NNN (LOGIC TEST AID)

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

B. STARTING SEQUENCE

TYPE "TNNN<CR>" WHERE "NNN" IS THE OCTAL ADDRESS OF THE "TSTX" TO BE EXECUTED. THE PROGRAM WILL THEN EXECUTE THE SUBTEST AND WILL REMAIN IN A SCOPE LOOP UNTIL THE COMPUTER IS STOPPED OF A <↑C> IS TYPED TO RETURN TO THE MONITOR.

C. RESTRICTIONS

SWITCH "11" MUST BE "0" [DOWN] TO RUN THIS TEST.

11. LISTING


```
                        %
                        .TITLE ADF11 PART I, LOGIC DIAGNOSTIC TEST
                        .ABS
                ;MAINDEC-11-DZADG-A-D
                ;COPYRIGHT JULY 17,1975
                ;DIGITAL EQUIPMENT CORP. MAYNARD MASS. 01754
                ;PROGRAMMER: EARL L. BOUSE
                ;            RAYMOND C. BALDWIN
                ;
                ;SWITCH REGISTER DEFINITIONS AND FUNCTIONS:

100000          SW15=100000                     ;=1, HALT ON ERROR
040000          SW14=40000                      ;=1, LOOP ON CURRENT TEST
020000          SW13=20000                      ;=1, SUPPRESS ERROR TYPEOUT
010000          SW12=10000                      ;=1, ISSUE 'RESET' IN ERROR LOOP
004000          SW11=4000                       ;=1, SUPPRESS 'SUBPROGRAM' ITERATIONS
002000          SW10=2000
```

```
                001000          SW09=1000
                000400          SW08=400
                000200          SW07=200
                000100          SW06=100
                000040          SW05=40
                000020          SW04=20
                000010          SW03=10
                000004          SW02=4
                000002          SW01=2
                000001          SW00=1

                                ;REGISTER DEFINITIONS

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

                                ;INSTRUCTIONS DEFINITIONS

                000003          ADDBR3=%3
                000004          ADCR4=%4
                000005          ADCSR5=%5
                005746          PUSH1SP=5746
                005726          POP1SP=5726
                024646          PUSH2SP=24646
                022626          POP2SP=22626
                000240          NOP=240
                000002          X=2
```

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

```
  402  000152  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  403  000154  000156                .+2
  404  000156  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  405  000160  000162                .+2
  406  000162  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  407  000164  000166                .+2
  408  000166  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  409  000170  000172                .+2
  410  000172  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  411  000174  000176                .+2
  412  000176  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  413  000200  000202                .+2
  414  000202  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  415  000204  000206                .+2
  416  000206  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  417  000210  000212                .+2
  418  000212  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  419  000214  000216                .+2
  420  000216  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  421  000220  000222                .+2
  422  000222  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  423  000224  000226                .+2
  424  000226  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  425  000230  000232                .+2
  426  000232  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  427  000234  000236                .+2
  428  000236  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  429  000240  000242                .+2
  430  000242  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  431  000244  000246                .+2
  432  000246  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  433  000250  000252                .+2
  434  000252  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  435  000254  000256                .+2
  436  000256  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  437  000260  000262                .+2
  438  000262  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  439  000264  000266                .+2
  440  000266  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  441  000270  000272                .+2
  442  000272  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  443  000274  000276                .+2
  444  000276  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  445  000300  000302                .+2
  446  000302  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  447  000304  000306                .+2
  448  000306  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  449  000310  000312                .+2
  450  000312  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  451  000314  000316                .+2
  452  000316  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  453  000320  000322                .+2
  454  000322  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  455  000324  000326                .+2
  456  000326  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  457  000330  000332                .+2
```

```
  458  000332  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  459  000334  000336          .+2
  460  000336  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  461  000340  000342          .+2
  462  000342  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  463  000344  000346          .+2
  464  000346  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  465  000350  000352          .+2
  466  000352  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  467  000354  000356          .+2
  468  000356  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  469  000360  000362          .+2
  470  000362  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  471  000364  000366          .+2
  472  000366  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  473  000370  000372          .+2
  474  000372  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  475  000374  000376          .+2
  476  000376  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  477  000400  000402          .+2
  478  000402  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  479  000404  000406          .+2
  480  000406  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  481  000410  000412          .+2
  482  000412  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  483  000414  000416          .+2
  484  000416  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  485  000420  000422          .+2
  486  000422  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  487  000424  000426          .+2
  488  000426  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  489  000430  000432          .+2
  490  000432  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  491  000434  000436          .+2
  492  000436  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  493  000440  000442          .+2
  494  000442  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  495  000444  000446          .+2
  496  000446  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  497  000450  000452          .+2
  498  000452  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  499  000454  000456          .+2
  500  000456  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  501  000460  000462          .+2
  502  000462  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  503  000464  000466          .+2
  504  000466  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  505  000470  000472          .+2
  506  000472  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  507  000474  000476          .+2
  508  000476  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  509  000500  000502          .+2
  510  000502  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  511  000504  000506          .+2
  512  000506  000004          4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
  513  000510  000512          .+2
```

```
514  000512  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
515  000514  000516                .+2
516  000516  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
517  000520  000522                .+2
518  000522  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
519  000524  000526                .+2
520  000526  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
521  000530  000532                .+2
522  000532  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
523  000534  000536                .+2
524  000536  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
525  000540  000542                .+2
526  000542  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
527  000544  000546                .+2
528  000546  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
529  000550  000552                .+2
530  000552  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
531  000554  000556                .+2
532  000556  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
533  000560  000562                .+2
534  000562  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
535  000564  000566                .+2
536  000566  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
537  000570  000572                .+2
538  000572  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
539  000574  000576                .+2
540  000576  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
541  000600  000602                .+2
542  000602  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
543  000604  000606                .+2
544  000606  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
545  000610  000612                .+2
546  000612  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
547  000614  000616                .+2
548  000616  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
549  000620  000622                .+2
550  000622  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
551  000624  000626                .+2
552  000626  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
553  000630  000632                .+2
554  000632  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
555  000634  000636                .+2
556  000636  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
557  000640  000642                .+2
558  000642  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
559  000644  000646                .+2
560  000646  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
561  000650  000652                .+2
562  000652  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
563  000654  000656                .+2
564  000656  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
565  000660  000662                .+2
566  000662  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
567  000664  000666                .+2
568  000666  000004                4                ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
569  000670  000672                .+2
```

```
570  000672  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
571  000674  000676                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
572  000676  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
573  000700  000702                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
574  000702  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
575  000704  000706                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
576  000706  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
577  000710  000712                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
578  000712  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
579  000714  000716                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
580  000716  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
581  000720  000722                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
582  000722  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
583  000724  000726                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
584  000726  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
585  000730  000732                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
586  000732  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
587  000734  000736                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
588  000736  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
589  000740  000742                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
590  000742  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
591  000744  000746                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
592  000746  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
593  000750  000752                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
594  000752  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
595  000754  000756                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
596  000756  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
597  000760  000762                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
598  000762  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
599  000764  000766                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
600  000766  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
601  000770  000772                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
602  000772  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
603  000774  000776                    4.+2                    ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
604  000776  000004                    4                       ;TRAPPED OR INTERRUPTED TO PREV. ADDR.
605
606           000020                   .=20
607  000020   011704                    ERTRAP                  ;ERROR TRAP REPORTING ROUTINE
608  000022   000340                    340
609  000024   011470                    PWRFAL                  ;POWER FAIL HANDLER
610  000026   000340                    340
611           000060                   .=60
612  000060   011020                    XTTYIN                  ;TELEPRINTER KEYBOARD ROUTINE
613  000062   000340                    340
614           000030                   .=30
615  000030   001200                    EMTSRV                  ;EMT TRAP, EMT DISPATCH SERVICE
616  000032   000340                    340
617  000034   012344                    LOGERR                  ;TRAP TRAP, LOGIC ERROR TRAP
618  000036   000340                    340
619           000174                   .=174
620  000174   000167  001132            JMP     INIT            ;INITIALIZATION ADDRESS
621  000200   000167  001376            JMP     MONITR          ;PROGRAM 'RESTART' ADDRESS
                     ;TRAP EQUIVALENCE TABLE:

           104400                     ERROR=TRAP                ;LOGIC TEST ERROR ROUTINE
           104000                     PRINT=EMT                 ;MESSAGE PRINTER ROUTINE
```

```
104001          DECOCT=EMT+1            ;DECIMAL TO OCTAL CONVERSIN ROUTINE
104002          SCOPE=EMT+2             ;LOGIC TEST SCOPE SUBROUTINE
104003          SPACE=EMT+3             ;TYPE 'N' SPACES
104004          PRTOCT=EMT+4            ;OCTAL PRINT ROUTINE
104005          TTYIN=EMT+5             ;TELETYPE INPUT ROUTINE
104006          TSTTKS=EMT+6            ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
104007          CKDONE=EMT+7            ;ROUTINE TO CHECK FOR 'A/D READY'
104010          NULL=EMT+10             ;ROUTINE TO PRINT NULL CHAR.'S
104011          INITAD=EMT+11           ;ROUTINE TO INITIALIZE THE A/D
104012          SAVREG=EMT+12           ;ROUTINE TO SAVE 'R1-R5' ON THE STACK
104013          GETREG=EMT+13           ;ROUTINE TO GET 'R1=R5' FROM THE STACK
```

```
                                        ;EMT DISPATCH SERVICE ROUTINE
                                        ;ARGUMENT OF EMT IS EXTRACTED AND USED AS OFFSET TO OBTAIN POINTER
                                        ;TO THE SELECTED SUBROUTINE.
            001200                              .=1200

001200  011646                 EMTSRV: MOV     (SP),-(SP)      ;GET PC FOR TO RETURN
001202  162716  000002                 SUB     #2,(SP)         ;PC OF EMT
001206  017616  000000                 MOV     @(SP),(SP)      ;GET EMT
001212  005716                         TST     (SP)            ;IS EMT VALID?
001214  001001                         BNE     EMTOK
001216  000000                         HALT                    ;INVALID EMT
001220  006316                 EMTOK:  ASL     (SP)            ;MULTIPLY EMT ARG BY '2'
001222  042716  177001                 BIC     #177001,(SP)    ;CLEAR UNWANTED BITS
001226  062716  001240                 ADD     #EMTTAB,(SP)    ;POINTER TO SUBROUTINE ADDRESS
001232  017616  000000                 MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
001236  000136                         JMP     @(SP)+          ;GO TO SUBROUTINE

                                        ;EMT DISPATCH TABLE

001240  012116                 EMTTAB: TYPMES                  ;MESSAGE PRINT ROUTINE
001242  011304                         BCDBIN                  ;DECIMAL TO BINARY CONVERSION ROUTINE
001244  012622                         SCOPEC                  ;LOGIC TEST SCOPE ROUTINE
001246  010724                         XSPACE                  ;SUBROUTINE TO TYPE SPACES
001250  012224                         OCTPRT                  ;OCTAL PRINT ROUTINE
001252  011020                         XTTYIN                  ;TELEPRINTER SERVICE ROUTINE
001254  012712                         TKSFLG                  ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
001256  011574                         XCKDONE                 ;ROUTINE TO TEST FOR 'A/D DONE'
001260  012724                         XNULL                   ;ROUTINE TO PRINT NULL CHAR.'S
001262  011756                         XINIT                   ;ROUTINE TO INITIALIZE THE A/D
001264  011766                         XSAVRG                  ;ROUTINE TO SAVE 'R1-R5' ON STACK
001266  012042                         XGETRG                  ;ROUTINE TO GET 'R1-R5' FROM STACK

                                ;REGISTER ADDRESSES
001270  177776                 PSW:    177776          ;ADDRESS OF PROCESSOR STATUS REG.
001272  177560                 TKS:    177560          ;ADDRESS OF KEYBOARD STATUS REG.
001274  177562                 TKB:    177562          ;   "    "    "      "    BUFFER   "
001276  177564                 TPS:    177564          ;   "    "  PRINTER STATUS REG.
001300  177566                 TPB:    177566          ;   "    "  PRINTER BUFFER REG.
001302  177570                 SWR:    177570          ;   "    "  SWITCH REG.
001304  177571                 SWR0:   177571          ;   "    "    "      "    HIGH BYTE
001306  164006                 ADCR:   164006          ;   "    "  A/D CONTROL REG.
001310  164010                 ADCSR:  164010          ;   "    "  A/D CONTROL & STATUS REG
001312  164004                 ADWCR:  164004          ;   "    "  A/D WORD COUNT REG.
001314  164000                 ADSWR:  164000          ;   "    "  A/D STATUS WORD REG.
001316  164012                 ADDBR:  164012          ;   "    "  A/D DATA BUFFER REG.
001320  164002                 ADWRA:  164002          ;   "    "  A/D WORD REG 'A'
001322  164014                 ADWRB:  164014          ;   "    "  A/D   "    "  'B'
001324  164016                 ADADR:  164016          ;   "    "  A/D OFFSET REG.
001326  000274                 ADINT:  0274            ;   "    "  A/D INTERRUPT VECTOR
001330  000276                 ADLVL:  0276            ;ADDRESS OF A/D INTERRUPT LEVEL
```

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

001332  016706  012444          INIT:   MOV     STACK,SP        ;INIT STACK POINTER=1000
001336  012777  000340  177724          MOV     #340,@PSW
001344  104000                        PRINT                   ;CALL MESSAGE PRINTER VIA 'EMT'
001346  012755                        TITLE                   ;TYPE PROGRAM HEADER.
001350  005067  012416          INIT1:  CLR     ADSIGN          ;UNIPOLAR=0,BIPOLAR=1
001354  104000                        PRINT
001356  013063                        MES2                    ;REQUEST THE A/D LENGTH
001360  104005                        TTYIN                   ;WAIT FOR ENTRY
001362  104001                        DECOCT                  ;CONVERT A/D LENGTH TO OCTAL
001364  012701  003776                MOV     #3776,R1        ;INIT AS 'INC MEM' OFFSET
001370  012702  001000                MOV     #1000,R2        ;= TO +5V VALUE FOR 10 BITS
001374  162767  000012  010056        SUB     #12,BCDTAB      ;A/D LENGTH = TO 10 BITS?
001402  001414                        BEQ     CORSIZ          ;YES, EXIT
001404  012703  000005                MOV     #5,R3           ;NO, TEST UP TO 15 BITS
001410  006301                SIZE:   ASL     R1              ;BUMP MEM. OFFSET
001412  006302                        ASL     R2              ;ALSO A/D SIZE
001414  005367  010040                DEC     BCDTAB          ;DECREMENT COUNT
001420  001405                        BEQ     CORSIZ          ;EXIT IF DONE
001422  005303                        DEC     R3
001424  100371                        BPL     SIZE            ;BRANCH UNTIL 15 IS REACHED
001426  104000                        PRINT                   ;ILLEGAL ENTRY
001430  013316                        QMARK                   ;PRINT '?'
001432  000746                        BR      INIT1           ;RETRY

001434  010167  012334        CORSIZ: MOV     R1,SIGNBF
001440  006267  012330                ASR     SIGNBF          ;SAVE A/D WORD LENGTH
001444  005167  012324                COM     SIGNBF          ;SET SIGN BITS
001450  005767  012316                TST     ADSIGN          ;TEST FOR SIGN BIT
001454  001401                        BEQ     .+4             ;BRANCH IF NOT SET
001456  006301                        ASL     R1              ;OTHERWISE ADD 1 BIT TO CONVERTER LENGTH.
001460  052701  000776                BIS     #776,R1         ;SET ALL POSSIBLE SHIFTED BITS
001464  012737  001532  000004        MOV     #INITA,@#4      ;INITIAL THE TIME OUT ADDRESS
001472  012737  000340  000006        MOV     #340,@#6
001500  005067  012314                CLR     INCFLG          ;CLR INCREMENT MEMORY FLAG
001504  062701  020000                ADD     #20000,R1       ;ADD 4K OFFSET TO A/D LENGTH
001510  010167  012306                MOV     R1,MEMSIZ       ;SAVE MEMORY SIZE
001514  006302                        ASL     R2              ;SET UP OFFSET FOR AVERAGING ROUTINE
001516  010267  012306                MOV     R2,ADSIZE       ;SAVE IT
001522  005737  037776        CORSZA: TST     @#37776         ;TEST IF 8K MEMORY IS AVAILABLE
001526  005267  012266                INC     INCFLG          ;SET SOFTWARE SWITCH
001532  012737  001546  000004 INITA:  MOV     #INITB,@#4      ;SET RET FOR FIRST MON EX MEM
001540  005001                        CLR     R1              ;TEST FOR MAX CORE IN SYSTEM
001542  005721                        TST     (R1)+           ;TRAP & RET TO INITA:
001544  000776                        BR      .-2
001546  005741                INITB:  TST     -(R1)           ;LAST CORE AVAILABLE
001550  162701  004000                SUB     #4000,R1
001554  010167  012244                MOV     R1,CORMAX       ;SAVE MAX CORE AVAILABLE
001560  012737  000006  000004        MOV     #6,@#4          ;RESTORE TIME OUT ADDRESS
001566  012737  000004  000006        MOV     #4,@#6          ;
```

```
001574  104000                          PRINT
001576  013172                          MES4                    ;PRINT THE TEST CALL LETTERS
001600  000411                          BR      INIT2           ;GO AND AWAIT COMMAND

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

001602  104010          MONITR: NULL
001604  000005                  RESET                           ;INITIALIZE ON ENTRY
001606  104010                  NULL
001610  016706  012166          MOV     STACK,SP                ;RESET STACK POINTER
001614  004767  010034          JSR     PC,CLRINT               ;CLR A/D INTR ADDR TO HALT
001620  104000                  PRINT                           ;CALL MESSAGE PRINTER
001622  013302                  CNTRLC                          ;TYPE '↑C'

001624  012767  001546  012152  INIT2:  MOV     #INITB,AVECTR   ;SET UP '↑A' VECTOR ADDRESS.
001632  104000                  PRINT
001634  013313                  DOT                             ;PRINT '.' TO INDICATE MONITOR READY
001636  104005                  TTYIN                           ;WAIT FOR TTY ENTRY
001640  122767  000104  007416  CMPB    #104,INBUF              ;TEST FOR 'D'
001646  001002                  BNE     .+6                     ;NOT 'D'
001650  000167  005770          JMP     DATA                    ;YES, RUN 'DATA' TEST
001654  122767  000114  007402  CMPB    #114,INBUF              ;TEST FOR 'L'
001662  001002                  BNE     .+6                     ;NOT 'L
001664  000167  000106          JMP     LOGIC                   ;YES, RUN 'LOGIC' TEST
001670  122767  000123  007366  CMPB    #123,INBUF              ;TEST FOR A "S"
001676  001002                  BNE     .+6                     ;NOT AN S
001700  000167  006100          JMP     SYNCO                   ;GO RUN SYNC TEST
001704  022767  000124  007352  CMP     #124,INBUF              ;TEST FOR SUBTEST
001712  001002                  BNE     .+6                     ;BRANCH IF NOT 'T'
001714  000167  006670          JMP     TESTX                   ;OTHERWISE RUN LOGIC SUBTEST
001720  104000          INIT3:  PRINT                           ;ILLEGAL ENTRY
001722  013316                  QMARK                           ;TYPE '?'
001724  000737                  BR      INIT2                   ;WAIT AGAIN
```

```
                                ;*****************************************************************
                                ;ADF11 LOGIC TEST
                                ;*****************************************************************

001726  104011          SETUP:  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
001730  016703  177362          MOV     ADDBR,ADDBR3    ;LOAD R3 WITH ADDBR ADDRESS
001734  016704  177346          MOV     ADCR,ADCR4      ;LOAD R4 WITH ADCR ADDRESS
001740  016705  177344          MOV     ADCSR,ADCSR5    ;LOAD R5 WITH ADCSR ADDRESS
001744  012767  002014  010736  MOV     #TST1,RETURN    ;SET UP RETURN ADDRESS FOR SCOPE
001752  012777  000340  177310  MOV     #340,@PSW       ;SET PROCESSOR PRIORITY TO '7'
001760  005067  010722          CLR     SCOPEF          ;CLR SOFTWARE FLAG
001764  005067  012036          CLR     SOFLAG          ;CLR SOFTWARE FLAG
001770  005067  012062          CLR     MESPRT          ;CLR SOFTWARE FLAG
001774  000207                  RTS     PC

001776  104000          LOGIC:  PRINT
002000  013321                  MESS                    ;TEXT LOGIC
002002  004767  177720  RESTRT: JSR     PC,SETUP        ;INITIALIZE LOGIC TEST
002006  012767  000030  010670  MOV     #30,ICOUNT      ;INITIALIZE LOGIC TEST

                                ;TEST THAT THE 'CSR' WAS INITIALIZED CORRECTLY

002014  000240          TST1:   NOP
002016  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'

002020  104002          TST2:   SCOPE
002022  005715                  TST     @ADCSR5
002024  001401                  BEQ     .+4
002026  104400                  ERROR

                                ;TEST THAT THE 'STATUS WORD REGISTER' WAS CLEARED VIA INIT

002030  104002          TST3:   SCOPE
002032  005777  177256          TST     @ADSWR
002036  001401                  BEQ     .+4
002040  104400                  ERROR

                                ;TEST THAT THE 'WORD REGISTER A' WAS CLEARED VIA INIT

002042  104002          TST4:   SCOPE
002044  005777  177250          TST     @ADWRA
002050  001401                  BEQ     .+4
002052  104400                  ERROR

                                ;TEST THAT THE 'WORD REGISTER B' WAS CLEARED VIA INIT

002054  104002          TST5:   SCOPE
002056  005777  177240          TST     @ADWRB
002062  001401                  BEQ     .+4
002064  104400                  ERROR
```

```
                                        ;WRITE THE 'DATA WORD REGISTER A' WITH 1'S

002066  104002                  TST6:   SCOPE
002070  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002072  012777  177777  177220          MOV     #177777,@ADWRA  ;WRITE ALL BITS
002100  022777  177777  177212          CMP     #177777,@ADWRA
002106  001401                          BEQ     .+4
002110  104400                          ERROR

                                        ;WRITE THE 'DATA WORD REGISTER B' WITH 1'S

002112  104002                  TST7:   SCOPE
002114  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002116  012777  177777  177176          MOV     #177777,@ADWRB
002124  022777  177777  177170          CMP     #177777,@ADWRB
002132  001401                          BEQ     .+4
002134  104400                          ERROR

                                        ;WRITE THE WORD COUNT REGISTER ALL 1'S

002136  104002                  TST10:  SCOPE
002140  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002142  012777  177777  177142          MOV     #177777,@ADWCR
002150  022777  177777  177134          CMP     #177777,@ADWCR
002156  001401                          BEQ     .+4
002160  104400                          ERROR

                                        ;WRITE THE STATUS WORD REGISTER ALL 1'S

002162  104002                  TST11:  SCOPE
002164  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002166  012777  177777  177120          MOV     #177777,@ADSWR
002174  022777  177777  177112          CMP     #177777,@ADSWR
002202  001401                          BEQ     .+4
002204  104400                          ERROR

                                        ;WRITE THE CONTROL & STATUS REGISTER WITH ALL 1'S EXCEPT 'GO' 'CLR FLAG' & INIT

002206  104002                  TST12:  SCOPE
002210  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002212  012715  153776                  MOV     #153776,@ADCSR5
002216  022715  151774                  CMP     #151774,@ADCSR5
002222  001401                          BEQ     .+4
002224  104400                          ERROR                   ;*BEWARE* BIT '6' IS CLEARED VIA ERROR SUBROUTINE
```

```
                                ;SELECT EXT SYN TO INHIBIT CONVERSION AND WRITE THE 'GO' BIT

002226  104002                  TST13:  SCOPE
002230  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002232  012714  004000                  MOV     #4000,@ADCR4    ;SET 'EXT, SYNC'
002236  012715  000001                  MOV     #1,@ADCSR5      ;SET 'GO'
002242  032715  000001                  BIT     #1,@ADCSR5      ;TEST IF BIT SET.
002246  001001                          BNE     .+4
002250  104400                          ERROR

                                ;TEST THAT THE 'GO' IS CLEARED IN P.C. MODE WHEN THE DATA BUFFER IS READ.

002252  104002                  TST14:  SCOPE
002254  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002256  012714  005000                  MOV     #5000,@ADCR4    ;SET FINAL CH. & EXT. SYNC
002262  005215                          INC     @ADCSR5         ;SET 'GO'
002264  005713                          TST     @ADDBR3         ;READ DATA
002266  032715  000001                  BIT     #1,@ADCSR5      ;TEST 'GO'
002272  001401                          BEQ     .+4             ;BRANCH IF CLR
002274  104400                          ERROR                   ;READING DATA BUFFER IN P.C. MODE DIDN'T C. 'GO'  .

                                ;TEST THAT 'A/D DONE' IS CLEARED WHEN DATA BUFFER IS READ

002276  104002                  TST15:  SCOPE
002300  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002302  052715  000200                  BIS     #200,@ADCSR5    ;SET DONE
002306  005713                          TST     @ADDBR3         ;READ DATA BUFFER
002310  105715                          TSTB    @ADCSR5         ;TEST DONE
002312  100001                          BPL     .+4             ;BRANCH IF CLR
002314  104400                          ERROR                   ;READING DATA BUFFER DIDN'T CLR DONE

                                ;SET XFER CHECK & WORDCOUNT CHECK (CSR BITS 3&8)
                                ;AND CHECK THAT BIT 2 SETS.

002316  104002                  TST16:  SCOPE
002320  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002322  052715  000410                  BIS     #410,@ADCSR5    ;SET WORDCOUNT CHECK AND XFER CHECK
002326  032715  000004                  BIT     #4,@ADCSR5              ;TEST DONE
002332  001001                          BNE     .+4             ;BRANCH IF SET
002334  104400                          ERROR                   ;SETTING W.C.CHECK & XFER CHECKDIDNT SET BIT 2

                                ;SELECT EXT SYNC TO INHIBIT CONVERSION AND WRITE THE INITAL CH.

002336  104002                  TST17:  SCOPE
002340  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002342  012714  004777                  MOV     #4777,@ADCR4
002346  022714  004777                  CMP     #4777,@ADCR4
002352  001401                          BEQ     .+4
002354  104400                          ERROR
```

```
                                ;TEST FOR WRITING THE 'CONTROL REG.' TO '0'

002356  104002                 TST20:  SCOPE
002360  104011                         INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002362  052715  004000                 BIS     #4000,@ADCSR5   ;CLR ALL A/D FLAGS
002366  012714  175777                 MOV     #175777,@ADCR4
002372  042714  176777                 BIC     #176777,@ADCR4  ;CLR ALL BITS EXCEPT FINAL
002376  022714  001000                 CMP     #1000,@ADCR4
002402  001401                         BEQ     .+4
002404  104400                         ERROR                   ;CAN'T CLR CONTROL REG.
002406  005713                         TST     @ADDBR3         ;CLR DONE

                                ;INHIBIT A CONVERSION VIA SETTING EXT SYNC. & WRITE THE 'FINAL' BIT TO '0'

002410  104002                 TST21:  SCOPE
002412  104011                         INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002414  052714  175777                 BIS     #175777,@ADCR4  ;WRITE ALL BITS BACK TO '1'
002420  042714  173777                 BIC     #173777,@ADCR4
002424  022714  004000                 CMP     #4000,@ADCR4
002430  001401                         BEQ     .+4
002432  104400                         ERROR
002434  005713                         TST     @ADDBR3         ;CLR DONE
```

```
                                ;TEST FOR WRITING THE CSR TO '0'

002436  104002                 TST22:  SCOPE
002440  104011                         INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002442  052714  004000                 BIS     #4000,@ADCR4            ;SET EXTERNAL SYNC
002446  052715  173777                 BIS     #173777,@ADCSR5         ;WRITE ALL BITS EXCEPT CLR FLAGS
002452  042715  173777                 BIC     #173777,@ADCSR5
002456  011501                         MOV     @ADCSR5,R1
002460  042701  100000                 BIC     #100000,R1              ;CLR ERROR BIT
002464  001401                         BEQ     .+4
002466  104400                         ERROR
002470  005713                         TST     @ADDBR3                 ;CLR DONE

                                ;SET ALL WRITABLE 'CSR' BITS AND TEST CLEARING THEM WITH 'CLR FLAGS'

002472  104002                 TST23:  SCOPE
002474  104011                         INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002476  052714  004000                 BIS     #4000,@ADCR4            ;SET EXT, SYNC.
002502  052715  153777                 BIS     #153777,@ADCSR5
002506  052715  004000                 BIS     #4000,@ADCSR5           ;ISSUE CLR FLAGS
002512  022715  000075                 CMP     #75,@ADCSR5             ;BITS '0,2,3,4,5' SHOULDN'T HAVE BEEN CLEARED
002516  001401                         BEQ     .+4
002520  104400                         ERROR                           ;CLR FLAG DIDN'T CLR ALL 'CSR' FLAGS
002522  005713                         TST     @ADDBR3                 ;CLR DONE

                                ;TEST FOR SETTING THE 'ERROR' BIT VIA FORCING 'CONVERSION IN PROCESS'

002524  104002                 TST24:  SCOPE
002526  104011                         INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002530  052715  040000                 BIS     #40000,@ADCSR5          ;SET 'CONV IN PRG'
002534  005715                         TST     @ADCSR5
002536  100401                         BMI     .+4
002540  104400                         ERROR                           ;CONV IN PROGRESS DIDN'T SET 'ERROR BIT'

                                ;TEST 'DATA OVERFLOW' FOR SETTING 'ERROR'

002542  104002                 TST25:  SCOPE
002544  104011                         INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002546  052715  010000                 BIS     #10000,@ADCSR5          ;SET 'DATA OVFLOW'
002552  005715                         TST     @ADCSR5
002554  100401                         BMI     .+4
002556  104400                         ERROR                           ;DATA OVFLOW DIDN'T SET 'ERROR BIT'
```

```
                                        ;TEST FOR SETTING THE 'ERROR' BIT IF IN P.C. SEQ. MODE W/ A/D DONE & FINAL SET

002560  104002                  TST26:  SCOPE
002562  104011                          INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002564  052714  101000                  BIS     #101000,@ADCSR5         ;SELECT: PC,, SEQ.
002570  052715  001200                  BIS     #1200,@ADCSR5           ;SET 'FINAL' & 'A/D DONE'
002574  005715                          TST     @ADCSR5                 ;TEST ERROR
002576  100401                          BMI     .+4                     ;BRANCH IF SET
002600  104400                          ERROR                           ;ERROR BIT DIDN'T SET FROM ABOVE SETUP
002602  005713                          TST     @ADDBR3                 ;CLR DONE

                                        ;TEST THAT STATUS WORD ADDRESS REGISTER CAN BE CLEARED

002604  104002                  TST27:  SCOPE
002606  104011                          INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
002610  052777  177777  176476        BIS     #177777,@ADSWR          ;SET IT
002616  005077  176472                CLR     @ADSWR                  ;CLR IT
002622  005777  176466                TST     @ADSWR
002626  001401                        BEQ     .+4
002630  104400                        ERROR                           ;CAN'T CLR A/D STATUS WORD REGISTER
                                        ;TEST THAT THE W.C. REGISTER CAN BE CLEARED

002632  104002                  TST30:  SCOPE
002634  052777  177777  176450        BIS     #177777,@ADWCR          ;SET IT
002642  005077  176444                CLR     @ADWCR                  ;CLR IT
002646  005777  176440                TST     @ADWCR
002652  001401                        BEQ     .+4
002654  104400                        ERROR                           ;CAN'T CLR A/D WORD COUNT REG.
002656  052777  177777  176426        BIS     #177777,@ADWCR
002664  052715  004000                BIS     #4000,@ADCSR5
002670  022777  177777  176414        CMP     #177777,@ADWCR
002676  001401                        BEQ     .+4
002700  104400                        ERROR                           ;ISSUING 'CLR FLAGS' ALTERED WORD COUNT REG.

                                        ;TEST THAT DATA WORD ADDRESS REGISTER 'A' CAN BE CLEARED

002702  104002                  TST31:  SCOPE
002704  052777  177777  176406        BIS     #177777,@ADWRA          ;SET IT
002712  005077  176402                CLR     @ADWRA                  ;CLEAR IT
002716  005777  176376                TST     @ADWRA
002722  001401                        BEQ     .+4
002724  104400                        ERROR                           ;CAN'T CLR WORD ADDRESS REG 'A'
002726  052777  177777  176364        BIS     #177777,@ADWRA
002734  052715  004000                BIS     #4000,@ADCSR5
002740  022777  177777  176352        CMP     #177777,@ADWRA
002746  001401                        BEQ     .+4
002750  104400                        ERROR                           ;ISSUING 'CLR FLAGS' ALTERED WORD REG. 'A'
```

```
                                ;TEST THAT DATA WORD ADDRESS REGISTER 'B' CAN BE CLEARED

002752  104002                  TST32:  SCOPE
002754  052777  177777  176340          BIS     #177777,@ADWRB
002762  005077  176334          CLR     @ADWRB
002766  005777  176330          TST     @ADWRB
002772  001401                  BEQ     .+4
002774  104400                  ERROR                   ;CAN'T CLR WORD ADDRESS REG. 'B'
002776  052777  177777  176316  BIS     #177777,@ADWRB
003004  052715  004000          BIS     #4000,@ADCSR5
003010  022777  177777  176304  CMP     #177777,@ADWRB
003016  001401                  BEQ     .+4
003020  104400                  ERROR                   ;ISSUING 'CLR FLAGS' ALTERED WORD REG. 'B'

                                ;*************************************************************************
                                ;AT THIS POINT ALL REGISTERS HAVE BEEN TESTED TO 'READ/WRITE'
                                ;THE NEXT SERIES OF TEST WILL TEST 'PROGRAM CONTROL' MODE
                                ;*************************************************************************

003022  104002                  TST33:  SCOPE
003024  012777  177777  176260  MOV     #-1,@ADWCR      ;LOAD W.C. '-1'
003032  004767  005720          JSR     PC,SETREG       ;LOAD  STATUS, WORD ADDRESSES A+B.
003036  012767  003044  007644  MOV     #TSTXB,RETURN   ;SET UP SCOPE ADDRESS
                                ;TEST THAT AN INTERRUPT OCCURRS IF READY IS SET

003044  000240                  TSTXB:  NOP             ;TEST STARTING ADDRESS

003046  052715  004000          BIS     #4000,@ADCSR5   ;CLR ALL FLAGS
003052  004767  006542          JSR     PC,LDINTR       ;LOAD INTERRUPT ADDRESS
003056  003106                  TAGAA
003060  012777  000140  176202  MOV     #140,@PSW       ;SET PROCESSOR PRIORITY @3
003066  052715  000300          BIS     #300,@ADCSR5    ;SET READY & INTR ENABLE
003072  016701  010700          MOV     DELAY1,R1
003076  005201                  INC     R1
003100  001376                  BNE     .-2
003102  104400                  ERROR                   ;READY DIDN'T CAUSE INTR @ PRIORITY 3
003104  000401                  BR      .+4
003106  022626                  TAGAA:  POP2SP          ;RESTORE STACK POINTER
003110  004767  006540          JSR     PC,CLRINT       ;CLR A/D INTR ADDRESS
```

```
                                        ;TEST THAT A/D DONE IS CLEARED ON THE INTERRUPT

003114  104002                 TST34:  SCOPE
003116  104011                         INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003120  004767  006474                 JSR     PC,LDINTR       ;LOAD INTERRUPT ADDRESS
003124  003154                         TAGAAA
003126  012777  000140  176134         MOV     #140,@PSW       ;SET PROCESSOR PRIORITY @3
003134  052715  000300                 BIS     #300,@ADCSR5    ;SET A/D DONE & INTR ENABLE
003140  016701  010632                 MOV     DELAY1,R1
003144  005201                         INC     R1
003146  001376                         BNE     .-2
003150  104400                         ERROR                   ;A/D DONE DIDN'T CAUSE INTR @ PRIORITY 3
003152  000401                         BR      .+4
003154  022626                 TAGAAA: POP2SP                  ;RESTORE STACK POINTER
003156  004767  006472                 JSR     PC,CLRINT       ;CLR A/D INTR ADDRESS
003162  105715                         TSTB    @ADCSR5         ;TEST DONE
003164  100001                         BPL     .+4             ;BRANCH IF CLR
003166  104400                         ERROR                   ;A/D DONE WASN'T CLEARED VIA INTERRUPT

                                        ;TEST THAT SETTING THE 'ERROR BIT' WILL CAUSE AN INTERRUPT

003170  104002                 TST35:  SCOPE
003172  104011                         INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003174  004767  006420                 JSR     PC,LDINTR       ;LOAD INTERRUPT ADDRESS
003200  003230                         TAGAB
003202  052777  000140  176060         BIS     #140,@PSW       ;SET PROC. PRIORITY @ 3
003210  012715  040100                 MOV     #40100,@ADCSR5  ;SET CONV. IN PROC. & INTR ENABLE
003214  016701  010556                 MOV     DELAY1,R1
003220  005201                         INC     R1
003222  001376                         BNE     .-2
003224  104400                         ERROR                   ;ERROR BIT DIDN'T CAUSE INTR
003226  000401                         BR      .+4
003230  022626                 TAGAB:  POP2SP                  ;RESET STACK POINTER
003232  004767  006416                 JSR     PC,CLRINT       ;CLR A/D INTR ADDRESS

                                        ;TEST THAT INTERRUPTS ARE INHIBITED WITH PROCESSOR PRIORITY @ 5

003236  104002                 TST36:  SCOPE
003240  104011                         INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003242  004767  006352                 JSR     PC,LDINTR       ;LOAD INTERRUPT ADDRESS
003246  003274                         TAGAC
003250  012777  000240  176012         MOV     #240,@PSW
003256  052715  000300                 BIS     #300,@ADCSR5    ;SET DONE & INTR ENABLE
003262  016701  010510                 MOV     DELAY1,R1
003266  005301                         DEC     R1
003270  001376                         BNE     .-2
003272  000402                         BR      .+6
003274  022626                 TAGAC:  POP2SP                  ;RESET STACK POINTER
003276  104400                         ERROR                   ;INTERRUPT OCCURRED @ PROC. PRIORITY '5'
003300  004767  006350                 JSR     PC,CLRINT       ;CLR A/D INTR ADDRESS
```

```
                                        ;TEST THAT INTERRUPTS ARE INHIBITED WITH PROCESSOR PRIORITY @ 4

003304  104002                  TST37:  SCOPE
003306  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003310  004767  006304                  JSR     PC,LDINTR       ;LOAD INTERRUPT ADDRESS
003314  003342                          TAGAD
003316  012777  000200  175744          MOV     #200,@PSW       ;SET PROC. PRIORITY @ 4
003324  052715  000300                  BIS     #300,@ADCSR5    ;SET A/D DONE & INTERRUPT ENABLE
003330  016701  010442                  MOV     DELAY1,R1
003334  005301                          DEC     R1
003336  001376                          BNE     .-2
003340  000402                          BR      .+6             ;OK, NO INTERRUPT OCCURRED
003342  022626                  TAGAD:  POP2SP                  ;RESET STACK
003344  104400                          ERROR                   ;INTERRUPT OCCURRED W/ PROC. @ PRIORITY '4'
003346  004767  006302                  JSR     PC,CLRINT       ;CLR A/D INTR ADDRESS

                                        ;TEST FOR SETTING 'XFER ERROR' VIA SETTING 'XFER CHECK .' & 'WORD COUNT OFLO'

003352  104002                  TST40:  SCOPE
003354  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003356  052715  000411                  BIS     #411,@ADCSR5    ;SET W.C & XFER CHECK &GO
003362  032715  000004                  BIT     #4,@ADCSR5      ;TEST 'XFER ERROR'
003366  001001                          BNE     .+4             ;BRANCH IF SET
003370  104400                          ERROR                   ;'XFER ERR' DIDN'T SET W/XFER CHF & W.C SET

                                        ;TEST 'A/D DONE' TO SET VIA TAKING P.C CONVERSION

003372  104002                  TST41:  SCOPE
003374  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003376  012714  001777                  MOV     #1777,@ADCR4    ;LOAD FINAL CH
003402  012714  000005                  MOV     #5,@ADCR4       ;SELECT PC, INITIAL CH. '5'
003406  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003410  104400                          ERROR                   ;A/D DONE FAILED TO SET (PC. MODE)
003412  005713                          TST     @ADDBR3         ;CLR DONE

                                        ;TEST 'A/D DONE' IS SET VIA STARTING A CONVERSION WITH 'GO'

003414  104002                  TST42:  SCOPE
003416  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003420  005215                          INC     @ADCSR5         ;SET 'GO'
003422  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003424  104400                          ERROR                   ;SETTING GO DIDN'T SET DONE (PC. MODE)
003426  005713                          TST     @ADDBR3         ;CLR DONE
```

```
                                ;TEST THAT LEAVING THE 'GO' BIT SET DOESN'T ALLOW CONTINUOUS CONVERSIONS

003430  104002                TST43:  SCOPE
003432  104011                        INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003434  005215                        INC     @ADCSR5         ;SET GO
003436  105715                        TSTB    @ADCSR5         ;WAIT FOR DONE
003440  100376                        BPL     .-2
003442  005777  175650                TST     @ADDBR          ;CLR DONE
003446  104007                        CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003450  000401                        BR      .+4             ;OK, DONE DIDN'T SET
003452  104400                        ERROR                   ;GO ENABLES CONTINUOUS P.C CONVERSIONS
003454  005713                        TST     @ADDBR3         ;CLR DONE

                                ;TEST THAT A CONVERSION IS INHIBITED IF 'XFER CHK' & 'W.C' & 'GO' ARE SET

003456  104002                TST44:  SCOPE
003460  104011                        INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003462  052714  001000                BIS     #1000,@ADCR4    ;SET FINAL TO INHIBIT 'GO'
003466  052715  000411                BIS     #411,@ADCSR5    ;SET W.C & XFER CHK & GO
003472  042715  000200                BIC     #200,@ADCSR5    ;CLR DONE
003476  012714  000010                MOV     #10,@ADCR4      ;START CONVERSION
003502  104007                        CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003504  000401                        BR      .+4             ;OK, DONE DIDN'T SET
003506  104400                        ERROR                   ;A/D DONE SET W/XFER CHK & W.C SET
003510  005713                        TST     @ADDBR3         ;CLR DONE

                                ;TEST THAT THE CONVERTER CONTINUES IF 'XFER CHK & W.C & GO' WERE SET, THEN CLR 'XFER'

003512  104002                TST45:  SCOPE
003514  104011                        INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003516  052714  001000                BIS     #1000,@ADCR4    ;SET FINAL TO INHIBIT 'GO'
003522  052715  000411                BIS     #411,@ADCSR5    ;SET XFER CHK & W.C. & GO
003526  042715  000200                BIC     #200,@ADCSR5    ;CLR DONE, (SET VIA SETTING W.C.)
003532  012714  000055                MOV     #55,@ADCR4      ;ISSUE FALSE START
003536  000240                        NOP
003540  000240                        NOP
003542  042715  000010                BIC     #10,@ADCSR5     ;CLR 'XFER CHK' (A/D SHOULD CONTINUE)

003546  104007                        CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003550  104400                        ERROR                   ;A/D DIDN'T CONTINUE AFTER CLR 'XFER CHK'
003552  005713                        TST     @ADDBR3         ;CLR DONE
```

```
                                ;TEST THAT CONVERSIONS ARE INHIBITED IF 'W.C' IS SET AND 'GO' IS CLEARED.

003554  104002                  TST46:  SCOPE
003556  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003560  052715  000400                  BIS     #400,@ADCSR5    ;SET 'W.C' FLAG, GO IS CLR
003564  042715  000200                  BIC     #200,@ADCSR5    ;CLR DONE
003570  012714  040075                  MOV     #40075,@ADCR4   ;START CONVERSION
003574  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003576  000401                          BR      .+4             ;OK, DONE DIDN'T SET
003600  104400                          ERROR                   ;A/D DONE SET W/'W.C' SET & 'GO' CLR.
003602  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'

                                ;TEST THAT SETTING 'GO' IN 'D M A' MODE DOESN'T ENABLE CONVERSIONS

003604  104002                  TST47:  SCOPE
003606  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003610  052715  000400                  BIS     #400,@ADCSR5    ;SET W.C. TO INHIBIT CONVRT
003614  042715  000200                  BIC     #200,@ADCSR5    ;CLR DONE, SET VIA SETTING W.C.
003620  012777  177777  175464          MOV     #-1,@ADWCR      ;PRECAUTIONARY DMA SETUP
003626  004767  005124                  JSR     PC,SETREG       ;SET UP A/D REG'S.
003632  012714  111000                  MOV     #111000,@ADCR4  ;SELECT; SEQ., DMA, FINAL CH.
003636  005215                          INC     @ADCSR5         ;SET 'GO'
003640  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003642  000401                          BR      .+4             ;OK, DONE DIDN'T SET
003644  104400                          ERROR                   ;SETTING 'GO' STARTED CONVRT IN 'DMA MODE'
003646  005713                          TST     @ADDBR3         ;CLR DONE
                                ;TEST THAT CONVERSIONS ARE INHIBITED WITH 'EXT. SYNC'. SELECTED

003650  104002                  TST50:  SCOPE
003652  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003654  012714  064000                  MOV     #64000,@ADCR4   ;SELECT P.C., EXT SYNC.
003660  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003662  000401                          BR      .+4             ;OK, DONE DIDN'T SET
003664  104400                          ERROR                   ;A/D DONE SET WITH 'EXT. SYN' SELECTED
003666  005713                          TST     @ADDBR3         ;CLR DONE
```

```
                                ;TEST 'A/D WAIT' VIA TAKING 4 CONSECUTIVE CONVERSIONS WITHOUT READING DATA.

003670  104002                  TST51:  SCOPE
003672  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
003674  012702  000004                  MOV     #4,R2
003700  012714  060007          TAGF:   MOV     #60007,@ADCR4   ;START CONVERSION
003704  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
003706  104400                          ERROR                   ;A/D DONE FAILED TO SET
003710  032715  002000                  BIT     #2000,@ADCSR5   ;TEST A/D WAIT
003714  001401                          BEQ     .+4             ;BRANCH IF NOT SET
003716  104400                          ERROR                   ;A/D WAIT SET BEFORE DATA BUFFER'S FULL
003720  042715  000200                  BIC     #200,@ADCSR5    ;CLR 'A/D DONE'
003724  005302                          DEC     R2              ;DEC CNTR.
003726  001364                          BNE     TAGF            ;BRANCH IF HAVEN'T TAKEN '4' CONVERSIONS

                                ;TEST THAT 'A/D WAIT' WILL SET ON THE 5TH CONVERSION AND INHIBIT FUTHER CONVERSIONS

003730  012714  060007          TAGG:   MOV     #60007,@ADCR4   ;START 5TH CONVERSION
003734  016701  010036                  MOV     DELAY1,R1
003740  005201                          INC     R1
003742  001376                          BNE     .-2

003744  032715  002000                  BIT     #2000,@ADCSR5   ;TEST A/D WAIT
003750  001001                          BNE     .+4             ;BRANCH IF SET
003752  104400                          ERROR                   ;5 CONSECT P.C. CONVERSIONS DIDN'T SET A/D WAIT

                                ;TEST THAT 'WAIT' INHIBITED 5TH CONVERSION

003754  105715                          TSTB    @ADCSR5         ;TEST DONE
003756  100001                          BPL     .+4             ;BRANCH IF CLR
003760  104400                          ERROR                   ;A/D DONE SET WITH A/D WAIT SET.

                                ;TEST THAT 'A/D WAIT' SET THE ERROR BIT

003762  005715                          TST     @ADCSR5         ;TEST ERROR
003764  100401                          BMI     .+4             ;BRANCH IF SET
003766  104400                          ERROR                   ;A/D WAIT DIDN'T SET ERROR BIT

                                ;TEST THAT 'A/D WAIT' CAN BE CLEARED VIA 'CLR FLAGS'.

003770  052715  004000                  BIS     #4000,@ADCSR5   ;CLR FLAGS
003774  032715  002000                  BIT     #2000,@ADCSR5   ;TEST A/D WAIT
004000  001401                          BEQ     .+4
004002  104400                          ERROR                   ;CLR FLAGS DIDN'T CLR 'A/D WAIT'
```

```
                                ;TEST THAT 'A/D WAIT' IS CLEARED VIA READING THE DATA BUFFER

004004  104002                 TST52:  SCOPE
004006  104011                         INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004010  012701  000005                 MOV     #5,R1           ;TAKE FIVE CONSECTIVE CONVERSIONS
004014  012714  060007         TAGXF:  MOV     #60007,@ADCR4   ;START CONVERSION
004020  016702  007752                 MOV     DELAY1,R2
004024  005202                         INC     R2              ;GIVE FLAG CHANCE TO SET
004026  001376                         BNE     .-2
004030  005301                         DEC     R1
004032  001370                         BNE     TAGXF
004034  032715  002000                 BIT     #2000,@ADCSR5   ;TEST A/D WAIT
004040  001001                         BNE     .+4             ;BRANCH IF SET
004042  104400                         ERROR                   ;A/D WAIT FAILED TO SET ON 5TH CONVERSION
004044  005713                         TST     @ADDBR3         ;READING DATA SHOULD ALLOW 5TH CONVERSION
004046  032715  002000                 BIT     #2000,@ADCSR5   ;RE-TEST 'WAIT
004052  001401                         BEQ     .+4
004054  104400                         ERROR                   ;READING DATA BUFFER DIDN'T CLR WAIT

                                ;TEST THAT A 5TH CONVERSION WILL BE TAKEN AFTER READING DATA

004056  016702  007714                 MOV     DELAY1,R2
004062  005202                         INC     R2              ;GIVE DONE A CHANCE TO SET
004064  001376                         BNE     .-2
004066  105715                         TSTB    @ADCSR5         ;TEST DONE
004070  100401                         BMI     .+4             ;BRANCH IF SET
004072  104400                         ERROR                   ;COULDN'T TAKE CONVERSION AFTER READING DATA
004074  005713                         TST     @ADDBR3         ;CLR DONE

                                ;TEST THE 'CR' UPDATE LOGIC VIA WRITTING '4' WORDS INTO THE 'CR'

004076  104002                 TST53:  SCOPE
004100  104011                         INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004102  012702  000004                 MOV     #4,R2           ;SET UP TO TAKE '4' CONVTS TO FILL DATA BUFFER
004106  005014                 TAGXL:  CLR     @ADCR4          ;ST. CONVERSION
004110  104007                         CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004112  104400                         ERROR                   ;A/D DONE DIDN'T SET
004114  042715  000200                 BIC     #200,@ADCSR5    ;CLR DONE
004120  005302                         DEC     R2
004122  001371                         BNE     TAGXL           ;ST. '4' CONVERSIONS

                                ;THE NEXT '4' LOADS SHOULD RE-LOAD THE 'CR' BUFFERS

004124  012714  000525                 MOV     #525,@ADCR4     ;SELECT; SEQ, SYNC, CH. '525'
004130  012714  000252                 MOV     #252,@ADCR4     ;SELECT; SEQ, SYNC, CH. '252'
004134  012714  000000                 MOV     #000,@ADCR4     ;SELECT; SEQ. SYNC, CH. '000'
004140  012714  000777                 MOV     #777,@ADCR4     ;SELECT; SEQ. SYNC, CH. '777'
```

```
                                ;ATTEMPT TO READ THE '4' STORED CONTROL WRD'S VIA UPDATING THE
                                ;CONTROL  BUFFER'S WITH A 'READ DATA' COMMAND

004144  022714  000525              CMP     #525,@ADCR4     ;SHOULD = 1ST WORD WRITTEN
004150  001401                      BEQ     .+4
004152  104400                      ERROR                   ;DATA IS NOT = TO 1ST WORD WRITTEN
004154  005713                      TST     @ADDBR3         ;READ DATA TO UPDATE BUFFER
004156  016702  007614              MOV     DELAY1,R2
004162  005302                      DEC     R2
004164  001376                      BNE     .-2
004166  022714  000252              CMP     #252,@ADCR4     ;SHOULD = 2ND WORD WRITTEN
004172  001401                      BEQ     .+4
004174  104400                      ERROR                   ;DATA IS NOT = TO 2ND WORD WRITTEN
004176  005713                      TST     @ADDBR3         ;READ DATA TO UPDATA BUFFER
004200  016702  007572              MOV     DELAY1,R2
004204  005302                      DEC     R2
004206  001376                      BNE     .-2
004210  005714                      TST     @ADCR4          ;SHOULD = 3RD WORD WRITTEN
004212  001401                      BEQ     .+4
004214  104400                      ERROR                   ;DATA IS NOT = TO 3RD WORD WRITTEN
004216  005713                      TST     @ADDBR3         ;READ DATA TO UPDATE BUFFER
004220  016702  007552              MOV     DELAY1,R2
004224  005302                      DEC     R2
004226  001376                      BNE     .-2
004230  022714  000777              CMP     #777,@ADCR4     ;SHOULD = 4TH WORD WRITTEN
004234  001401                      BEQ     .+4
004236  104400                      ERROR                   ;DATA IS NOT = TO 4TH WORD WRITTEN

                                ;ATTEMPT TO WRITE '5' WORDS  INTO THE CONTROL REG. WITHOUT READING
                                ;DATA AND TEST THAT THE '5TH' WORD IS LOCKED OUT AND THAT IT DOESN'T
                                ;WRITE OVER THE '4TH' WORD WRITTEN.

004240  104002                  TST54:  SCOPE
004242  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004244  012702  000004                  MOV     #4,R2           ;SET UP TO TAKE '4' CONVTS TO FILL DATA BUFFER
004250  005014                  TAGXX:  CLR     @ADCR4          ;ST. CONVERSION
004252  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004254  104400                          ERROR                   ;A/D DONE DIDN'T SET
004256  042715  000200                  BIC     #200,@ADCSR5    ;CLR DONE
004262  005302                          DEC     R2
004264  001371                          BNE     TAGXX           ;ST. '4' CONVERSIONS
004266  012714  000525                  MOV     #525,@ADCR4     ;SELECT; CH. '525'
004272  012714  000252                  MOV     #252,@ADCR4     ;SELECT; CH. '252'
004276  005014                          CLR     @ADCR4          ;SELECT; CH. '000'
004300  012714  000777                  MOV     #777,@ADCR4     ;SELECT; CH. '777'
004304  005014                          CLR     @ADCR4          ;SELECT; CH. '000'
```

```
                                ;ALL 5 WORDS HAVE BEEN WRITTEN
004306  012700  000003                MOV     #3,R0
004312  005713          TAGXM:  TST     @ADDBR3         ;ISSUE 3 READS TO UPDATE BUFFER
004314  016702  007456          MOV     DELAY1,R2
004320  005302                  DEC     R2
004322  001376                  BNE     .-2
004324  005300                  DEC     R0
004326  001371                  BNE     TAGXM
004330  022714  000777          CMP     #777,@ADCR4     ;SHOULD = 4TH WORD WRITTEN
004334  001401                  BEQ     .+4
004336  104400                  ERROR                   ;DATA IS NOT = TO 4TH WORD WRITTEN

                                ;TEST FOR TAKING A CONVERSION WITH SINGLE CH. SELECTED.

004340  104002          TST55:  SCOPE
004342  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004344  012714  101027          MOV     #101027,@ADCR4  ;SELECT FINAL
004350  012714  160027          MOV     #160027,@ADCR4  ;LOAD INITIAL & ST.
004354  104007                  CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004356  104400                  ERROR                   ;SINGLE CH. SELECTED, DIDN'T SET DONE
004360  005713                  TST     @ADDBR3         ;CLR DONE

                                ;TEST THAT 'FINAL CH.' FLAG WAS NOT SET WITH SINGLE CH. SELECTED

004362  032715  001000          BIT     #1000,@ADCSR5   ;TEST FINAL CH.
004366  001401                  BEQ     .+4
004370  104400                  ERROR                   ;SINGLE CH SET FINAL CH. FLAG.

                                ;TEST THAT INITIAL CH. DOESNT GET INCREMENTED IN SINGLE CH. MODE

004372  104002          TST56:  SCOPE
004374  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004376  012714  101077          MOV     #101077,@ADCR4  ;LOAD FINAL
004402  012714  100077          MOV     #100077,@ADCR4  ;LOAD INITAL CH.
004406  104007                  CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004410  104400                  ERROR                   ;A/D DONE FAILED TO SET
004412  005713                  TST     @ADDBR3         ;CLR DONE
004414  005215                  INC     @ADCSR5         ;SET 'GO', ST 2ND CONVERSION
004416  104007                  CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004420  104400                  ERROR                   ;A/D DONE FAILED TO SET
004422  005713                  TST     @ADDBR3         ;CLR DONE
004424  022714  100077          CMP     #100077,@ADCR4  ;CH. CHANGED?
004430  001401                  BEQ     .+4
004432  104400                  ERROR                   ;INITIAL CH. REG WAS UPDATED IN SINGLE CH
```

```
                                ;TEST THAT THE FINAL CH. FLAG IS SET WHEN INITIAL = FINAL

004434  104002                  TST57:  SCOPE
004436  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004440  012714  101001                  MOV     #101001,@ADCR4  ;LOAD FINAL CH. =1
004444  012714  100000                  MOV     #100000,@ADCR4  ;LOAD INITIAL CH.=0
004450  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004452  104400                          ERROR                   ;DONE FAILED TO SET
004454  005713                          TST     @ADDBR3         ;CLR DONE
004456  032715  001000                  BIT     #1000,@ADCSR5   ;TEST FINAL CH. FLAG
004462  001401                          BEQ     .+4             ;CONTINUE IF NOT SET
004464  104400                          ERROR                   ;FINAL CH. FLAG SET BEFORE FINAL CH. WAS REACHED
004466  005215                          INC     @ADCSR5         ;ST 2RD CONVERSION
004470  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004472  104400                          ERROR                   ;DONE FAILED TO SET
004474  032715  001000                  BIT     #1000,@ADCSR5   ;TEST FINAL CH.
004500  001001                          BNE     .+4             ;BR IF SET
004502  104400                          ERROR                   ;FINAL CH. FLAG DIDN'T SET W/ INITIAL = FINAL
004504  005713                          TST     @ADDBR3         ;CLR DONE

                                ;TEST THAT INITIAL CH. ADDRESS GETS UPDATED AFTER EACH NON SINGLE CH. CONVERSION

004506  104002                  TST60:  SCOPE
004510  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004512  012714  001777                  MOV     #1777,@ADCR4    ;LOAD FINAL CH: 777
004516  012714  100000                  MOV     #100000,@ADCR4  ;LOAD INITIAL  CH: '0'
004522  012702  000776                  MOV     #776,R2         ;SET UP TO '776' CONVERSIONS
004526  104007          TAGL:   CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004530  104400                  ERROR                   ;A/D DONE FAILED TO SET
004532  005713                  TST     @ADDBR3         ;CLR DONE
004534  005302                  DEC     R2
004536  001402                  BEQ     TAGM            ;BRANCH IF TAKEN ALL CONVERSIONS
004540  005215                  INC     @ADCSR5         ;OTHERWISE START NEXT CONVERSION
004542  000771                  BR      TAGL
```

```
004544  011501          TAGM:   MOV     @ADCSR5,R1      ;SAVE CONTENTS OF 'CSR'
004546  011402                  MOV     @ADCR4,R2
004550  042702  177000          BIC     #177000,R2      ;CLR ALL BUT CH. BITS
004554  022702  000776          CMP     #776,R2         ;DOES INITIAL = FINAL
004560  001401                  BEQ     .+4             ;BRANCH IF YES
004562  104400                  ERROR                   ;INITIAL CH. REG WASN'T INCREMENTED

                        ;TEST THAT THE 'INITIAL CH. REG. IS RESET AFTER REACHING THE FINAL CH.

004564  104002          TST61:  SCOPE
004566  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004570  012714  001007          MOV     #1007,@ADCR4    ;SET FINAL CH. = 7
004574  012714  160005          MOV     #160005,@ADCR4  ;SET INITIAL CH = 5
004600  012702  000003          MOV     #3,R2
004604  104007          TAGO:   CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004606  104400                  ERROR                   ;A/D DONE FAILED TO SET
004610  005713                  TST     @ADDBR3         ;CLR 'GO' BIT
004612  005302                  DEC     R2
004614  001402                  BEQ     TAGP
004616  005215                  INC     @ADCSR5         ;SET 'GO'
004620  000771                  BR      TAGO

004622  011401          TAGP:   MOV     @ADCR4,R1
004624  042701  177000          BIC     #177000,R1      ;CLR UNWANTED BITS
004630  022701  000005          CMP     #5,R1           ;SHOULD OF RESET TO '5'
004634  001401                  BEQ     .+4             ;BRANCH IF YES
004636  104400                  ERROR                   ;INITIAL CH. REG. WASN'T RESET AFTER REACHING FINAL

                        ;TEST THAT 'INITIAL CH.' REG IS NOT INCREMENTED IN RANDOM MODE

004640  104002          TST62:  SCOPE
004642  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004644  012714  060005          MOV     #60005,@ADCR4   ;LOAD INITAL CH.
004650  104007                  CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004652  104400                  ERROR                   ;A/D DONE FAILED TO SET
004654  005713                  TST     @ADDBR3         ;CLR DONE
004656  011401                  MOV     @ADCR4,R1
004660  042701  177000          BIC     #177000,R1      ;CLR ALL BUT CH. BITS
004664  022701  000005          CMP     #5,R1           ;CH. STILL SHOULD EQUAL '5'
004670  001401                  BEQ     .+4             ;BRANCH IF YES
004672  104400                  ERROR                   ;INITIAL CH. REG WAS UPDATED IN RANDOM MODE
```

```
                                ;*************************************************************************
                                ;TEST THE 'SEQUENTIAL DMA' MODE WITH WITH 'XFER CHK' SET TO STOP ON WC.
                                ;*************************************************************************

004674  104002                  TST63:  SCOPE
004676  104011                          INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004700  012777  177777  174404          MOV     #-1,@ADWCR              ;SET UP TO TAKE '1' CONVERSIONS
004706  004767  004044          JSR     PC,SETREG               ;LOAD A/D REG.S
004712  012714  101000          MOV     #101000,@ADCR4          ;LOAD FINAL CH.=0
004716  012714  110000          MOV     #110000,@ADCR4          ;SELECT: SEQ., DMA, CH.0
004722  104007                  CKDONE                          ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004724  104400                  ERROR                           ;W.C. OVRFLO DIDN'T SET DONE IN SEQ. MODE

                                ;TEST THAT THE 'W.C.' REG. IS RESET TO '-1' ON THE W.C. OVFLO

004726  104002                  TST64:  SCOPE
004730  104011                          INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
004732  012777  177777  174352          MOV     #-1,@ADWCR              ;SET UP TO TAKE '1' CONVERSIONS
004740  004767  004012          JSR     PC,SETREG               ;LOAD A/D REG.S
004744  012715  000010          MOV     #10,@ADCSR5             ;SET 'XFER CHK' TO HALT ON WC. OVRFLO
004750  012714  001007          MOV     #1007,@ADCR4            ;LOAD FINAL CH.=7
004754  012714  110000          MOV     #110000,@ADCR4          ;SELECT: SEQ., DMA, CH.0
004760  104007                  CKDONE                          ;SUBROUTINE TO CHECK FOR 'A/D DONE'
004762  104400                  ERROR                           ;W.C. OVRFLO DIDN'T SET DONE IN SEQ. MODE
004764  022777  177777  174320          CMP     #-1,@ADWCR              ;WAS THE 'W.C.' REG RESET
004772  001401                  BEQ     .+4
004774  104400                  ERROR                           ;W.C. OVRFLO DIDN'T RESET W.C. REG.
                                ;TEST THAT 'DATA WORD REG. A' IS RESET ON W.C. OVRFLO

004776  104002                  TST65:  SCOPE
005000  104011                          INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
005002  012777  177777  174302          MOV     #-1,@ADWCR              ;SET UP TO TAKE '1' CONVERSIONS
005010  004767  003742          JSR     PC,SETREG               ;LOAD A/D REG.S
005014  012715  000010          MOV     #10,@ADCSR5             ;SET 'XFER CHK' TO HALT ON WC. OVRFLO
005020  012714  001007          MOV     #1007,@ADCR4            ;LOAD FINAL CH.=7
005024  012714  110000          MOV     #110000,@ADCR4          ;SELECT: SEQ., DMA, CH.0
005030  104007                  CKDONE                          ;SUBROUTINE TO CHECK FOR 'A/D DONE'
005032  104400                  ERROR                           ;W.C. OVRFLO DIDN'T SET DONE IN SEQ. MODE
005034  022777  015072  174256          CMP     #ADBUFF,@ADWRA          ;WAS WORD REG. A RESET
005042  001401                  BEQ     .+4
005044  104400                  ERROR                           ;DATA WRD REG. 'A' WASN'T RESET ON 'W.C.' OVRFLO
```

```
                                        ;TEST THAT 'DATA WORD REG. B' IS RESET ON W.C. OVRFLO

005046  104002                  TST66:  SCOPE
005050  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
005052  012777  177777  174232          MOV     #-1,@ADWCR      ;SET UP TO TAKE '1' CONVERSIONS
005060  004767  003672                  JSR     PC,SETREG       ;LOAD A/D REG.S
005064  012715  000010                  MOV     #10,@ADCSR5     ;SET 'XFER CHK' TO HALT ON WC. OVRFLO
005070  012714  001007                  MOV     #1007,@ADCR4    ;LOAD FINAL CH.=7
005074  012714  110000                  MOV     #110000,@ADCR4  ;SELECT: SEQ., DMA, CH.0
005100  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
005102  104400                          ERROR                   ;W.C. OVRFLO DIDN'T SET DONE IN SEQ. MODE
005104  022777  015074  174210          CMP     #ADBUFF+2,@ADWRB        ;WAS WORD REG. B RESET
005112  001401                          BEQ     .+4
005114  104400                          ERROR                   ;DATA WRD REG. 'B' WASN'T RESET ON 'W.C.' OVRFLO

                                        ;TEST THAT THE 'STATUS WORD REG.' IS RESET ON W.C. OVRFLO

005116  104002                  TST67:  SCOPE
005120  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
005122  012777  177777  174162          MOV     #-1,@ADWCR      ;SET UP TO TAKE '1' CONVERSIONS
005130  004767  003622                  JSR     PC,SETREG       ;LOAD A/D REG.S
005134  012715  000010                  MOV     #10,@ADCSR5     ;SET 'XFER CHK' TO HALT ON WC. OVRFLO
005140  012714  001007                  MOV     #1007,@ADCR4    ;LOAD FINAL CH.=7
005144  012714  110000                  MOV     #110000,@ADCR4  ;SELECT: SEQ., DMA, CH.0
005150  104007                          CKDONE                  ;SUBROUTIN TO CHECK FOR 'A/D DONE'
005152  104400                          ERROR                   ;W.C. OVRFLO DIDN'T SET DONE IN SEQ. MODE
005154  022777  014070  174132          CMP     #RANBUF,@ADSWR  ;WAS STATUS REG. RESET++++
005162  001401                          BEQ     .+4
005164  104400                          ERROR                   ;STATUS WRDREG. WASN'T RESET VIA W.C. OVRFLD

                                        ;TEST THAT DATA WORD ADDRESS 'A' GETS MODIFIED VIA THE 'DMA' TRANSFER

005166  104002                  TST70:  SCOPE
005170  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
005172  012777  177770  174112          MOV     #-10,@ADWCR     ;SET UP TO TAKE '8' CONVERSIONS
005200  004767  003552                  JSR     PC,SETREG       ;LOAD A/D REG.S
005204  012700  177760                  MOV     #-20,R0
005210  012702  015072                  MOV     #ADBUFF,R2      ;PRE-LOAD DATA BUFFERS 'A&B'
005214  012722  125252          TAGAF:  MOV     #125252,(R2)+
005220  005200                          INC     R0
005222  001374                          BNE     TAGAF
005224  012715  000010                  MOV     #10,@ADCSR5     ;SET 'XFER CHK' TO HALT ON WC. OVRFLO
005230  012714  001007                  MOV     #1007,@ADCR4    ;LOAD FINAL CH.=7
005234  012714  110000                  MOV     #110000,@ADCR4  ;SELECT: SEQ., DMA, CH.0
005240  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
005242  104400                          ERROR                   ;W.C. OVRFLO DIDN'T SET IN SEQ. MODE
```

```
                                ;CHECK THAT DATA BUFFER 'A' WAS MODIFIED VIA DMA

005244  012700  177770                MOV     #-10,R0
005250  012701  015072                MOV     #ADBUFF,R1
005254  022721  125252        TAGAG:  CMP     #125252,(R1)+   ;CHECK IF ADDRESS WAS MODIFED
005260  001002                BNE     .+6
005262  104400                ERROR                   ;MEMORY WASN'T MODIFIED VIA SEQ. DMA
005264  000402                BR      .+6             ;EXIT ON ERROR
005266  005200                INC     R0
005270  001371                BNE     TAGAG

                                ;TEST THAT ONLY '8' LOCATIONS WERE MODIFIED WITH WC.=-10 &'XFER CHK' SET

005272  012700  177770                MOV     #-10,R0
005276  012701  015112                MOV     #ADBUFF+20,R1
005302  022721  125252        TAGAH:  CMP     #125252,(R1)+   ;CHECK BUFFER 'B'
005306  001402                BEQ     .+6             ;BRANCH IF UNMODIFIED
005310  104400                ERROR                   ;DATA BUFFER 'B' WAS MODIFIED W/ 'XFER CHK' SET
005312  000402                BR      .+6             ;EXIT ON ERROR
005314  005200                INC     R0
005316  001371                BNE     TAGAH
```

```
                                ;TEST THAT BOTH BUFFERS 'A&B' ARE MODIFIED IF 'XFER CHK' IS CLR & 'GO' IS SET

005320  104002                  TST71:  SCOPE
005322  104011                          INITAD                  ;CALL ROUTIME TO INITIALIZE THE 'A/D'
005324  012777  177700  173760          MOV     #-100,@ADWCR    ;SET UP FOR '200' CONVERSION; A-100,B-100
005332  004767  003420                  JSR     PC,SETREG       ;LOAD A/D REG.S
005336  062777  000100  173756          ADD     #100,@ADWRB     ;OFFSET 'B' 200 LOCATION FROM 'A'
005344  012701  015072                  MOV     #ADBUFF,R1      ;THIS WILL GIVE A '100' BYTE (32 WRD) GAP
005350  012700  177500                  MOV     #-300,R0        ;BETWEEN THE END OF 'A' & ST. OF 'B'
005354  012721  152525          TAGAI:  MOV     #152525,(R1)+   ;PRE-LOAD BUFFER AREA
005360  005200                          INC     R0
005362  001374                          BNE     TAGAI
005364  004767  004230                  JSR     PC,LDINTR       ;LOAD A/D INTR. ADDR.
005370  005424                          TAGAJ                   ;INTR HERE
005372  012714  001777                  MOV     #1777,@ADCR4    ;LOAD FINAL CH.
005376  005215                          INC     @ADCSR5         ;SET 'GO'
005400  012714  110000                  MOV     #110000,@ADCR4  ;SELECT; SEQ.,DMA, INITIAL CH.
005404  012700  177740                  MOV     #-40,R0         ;LET A/D COMPLETE A FEW CONVERSIONS
005410  005200                          INC     R0              ;THEN RELOAD BUFFER 'A'
005412  001376                          BNE     .-2             ;TO CHK THAT IT IS NOT RE-MODIFIED
005414  012767  125252  007450          MOV     #125252,ADBUFF  ;RELOAD 'A' BUFFER WITH KNOWN VALUE
005422  000001                          WAIT                    ;WAIT FOR INTERRUPT

                                ;ENTER HERE ON INTERRUPT

005424  052715  004000          TAGAJ:  BIS     #4000,@ADCSR5   ;CLR ALL FLAGS
005430  105715                          TSTB    @ADCSR5         ;TEST FOR 2ND W.C. OVRFLO
005432  100376                          BPL     .-2
005434  004767  004214                  JSR     PC,CLRINT       ;CLR OUT INTERRUPT
005440  022626                          POP2SP                  ;RESET STACK POINTER

                                ;TEST THAT BUFFER 'A' WASN'T REMODIFIED AFTER CLEARING 'GO'

005442  022767  125252  007422          CMP     #125252,ADBUFF  ;EXAMINE BUFFER
005450  001401                          BEQ     .+4             ;BRANCH IF NOT MODIFIED
005452  104400                          ERROR                   ;'STOP' DIDN'T STOP A/D

                                ;TEST THAT THE '32' WRD GAP WASN'T MODIFIED

005454  012700  177741                  MOV     #-37,R0
005460  012701  015272                  MOV     #ADBUFF+200,R1  ;LOAD ST. ADDR OF GAP
005464  022721  152525          TAGXJ:  CMP     #152525,(R1)+   ;TEST LOCATION
005470  001402                          BEQ     .+6
005472  104400                          ERROR                   ;THE '32' WRD GAP AFTER WRD 'A' WAS MODIFIED
005474  000402                          BR      .+6             ;EXIT ON ERROR
005476  005200                          INC     R0
005500  001371                          BNE     TAGXJ
```

```
                                        ;TEST THAT BUFFER 'B' W A S  MODIFIED

005502  012700  000100                        MOV     #100,R0
005506  012701  015372                        MOV     #ADBUFF+300,R1  ;LOAD BUFFER 'B' STARTING ADDRESS
005512  022721  152525        TAGAK:  CMP     #152525,(R1)+   ;WAS ADDRESS MODIFIED?
005516  001002                        BNE     .+6             ;BRANCH IF YES
005520  104400                        ERROR                   ;BUFFER 'B' WASN'T MODIFIED
005522  000402                        BR      .+6             ;EXIT ON ERROR
005524  005300                        DEC     R0
005526  001371                        BNE     TAGAK

                                        ;TEST THAT THE INITIAL CH. REG. CONTAINS  '177' AFTER TAKING '200' CONVERSIONS

005530  104002        TST72:  SCOPE
005532  104011                        INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
005534  012777  177700  173550        MOV     #-100,@ADWCR    ;SET UP FOR '200' CONVERSION; A-100,B-100
005542  004767  003210                JSR     PC,SETREG       ;LOAD A/D REG
005546  004767  004046                JSR     PC,LDINTR       ;LOAD A/D INTR. ADDR.
005552  005570                        TAGYK                   ;INTR HERE
005554  012714  001777                MOV     #1777,@ADCR4    ;LOAD FINAL CH.
005560  005215                        INC     @ADCSR5         ;SET 'GO'
005562  012714  110000                MOV     #110000,@ADCR4  ;SELECT; SEQ., DMA, INITIAL CH.
005566  000001                        WAIT                    ;WAIT FOR INTERRUPT

                                        ;ENTER HERE ON INTERRUPT

005570  052715  004000        TAGYK:  BIS     #4000,@ADCSR5   ;CLR ALL A/D FLAGS
005574  022626                        POP2SP                  ;RESET STACK POINTER
005576  004767  004052                JSR     PC,CLRINT       ;CLR OUT INTR ADDRESS
005602  105715                        TSTB    @ADCSR5         ;WAIT FOR 2ND W.C OVRFLO
005604  001776                        BEQ     .-2
005606  022714  110177                CMP     #110177,@ADCR4  ;READ THE 'CR'
005612  001401                        BEQ     .+4
005614  104400                        ERROR                   ;'CR' DOSEN'T CONTAIN FINAL CH.

                                        ;*************************************************************************
                                        ;NOW IT HAS BEEN CONFIRMED THAT WE CAN DO D.M.A. TRANSFERS, ATTEMPT A
                                        ;'D.M.A.' TRANSFER TO A NON EXISTANT MEMORY LOCATION TO TEST THAT THE
                                        ;A/D WILL RELEASE THE BUS AND SET THE 'NXM' FLAG.
                                        ;*************************************************************************

005616  104002        TST73:  SCOPE
005620  104011                        INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
005622  012777  177766  173462        MOV     #-10.,@ADWCR    ;SET UP FOR 8 CONVERSIONS
005630  004767  003122                JSR     PC,SETREG       ;SET UP A/D REG.
005634  052715  000060                BIS     #60,@ADCSR5     ;SET EA BITS '16 & 17'
005640  012777  173000  173452        MOV     #173000,@ADWRA  ;SET WORD ADDR 'A' TO AN ILLEGAL ADDR.
005646  004767  003746                JSR     PC,LDINTR       ;LOAD INTERRUPT VECTOR ADDRESS
005652  005666                        TAGAL
005654  012714  001000                MOV     #1000,@ADCR4    ;LOAD FINAL CH. OF '0'
005660  012714  110000                MOV     #110000,@ADCR4  ;LOAD INITIAL CH. & START
005664  000001                        WAIT                    ;WILL IT EVER COME BACK?
```

```
                                        ;ENTER HERE ON INTERRUPT

005666  022626                  TAGAL:  POP2SP                  ;RESET STACK POINTER
005670  004767  003760                  JSR     PC,CLRINT       ;RESTORE INTERRUPT ADDRESS
005674  005715                          TST     @ADCSR5         ;TEST FOR ERROR BIT
005676  100401                          BMI     .+4             ;BRANCH IF SET
005700  104400                          ERROR                   ;NMX DIDN'T SET 'ERROR' BIT

                                        ;*************************************************************************
                                        ;TEST THAT A 'D.M.A' RANDOM MODE TRANSFER CAN BE MADE.
                                        ;*************************************************************************

005702  104002                  TST74:  SCOPE
005704  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
005706  012701  030000                  MOV     #30000,R1       ;SELECT: RAN,DMA,CH. '0'
005712  012702  014070                  MOV     #RANBUF,R2      ;SETUP TO LOAD STATUS WORD BUFFER
005716  010122                  TAGYN:  MOV     R1,(R2)+        ;LOAD CH.S '0-310'
005720  005201                          INC     R1
005722  022701  030311                  CMP     #30311,R1       ;LOADED ALL CH.S?
005726  001373                          BNE     TAGYN           ;BRANCH IF NO

005730  012777  177777  173354          MOV     #-1,@ADWCR      ;LOAD W.C
005736  004767  003014                  JSR     PC,SETREG       ;LOAD A/D REGS.
005742  012714  030125                  MOV     #30125,@ADCR4   ;START CONVERSION
005746  104007                          CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
005750  104400                          ERROR                   ;RANDOM DIDN'T SET DONE
```

```
                        ;*************************************************************************
                        ;TEST THAT THE CORRECT STATUS WORD IS LOADED FROM THE STATUS BUFFER
                        ;*************************************************************************

005752  104002          TST75:  SCOPE
005754  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
005756  012701  030000          MOV     #30000,R1       ;SELECT; RAN,DMA,CH. '0'
005762  012702  014070          MOV     #RANBUF,R2      ;SETUP TO LOAD STATUS WORD BUFFER
005766  010122          TAGXN:  MOV     R1,(R2)+        ;LOAD CH.S '0-310'
005770  005201                  INC     R1
005772  022701  030311          CMP     #30311,R1       ;LOADED ALL CH.S?
005776  001373                  BNE     TAGXN           ;BRANCH IF NO
006000  012777  177777  173304  MOV     #-1,@ADWCR      ;LOAD W.C
006006  004767  002744          JSR     PC,SETREG       ;LOAD A/D REGS.
006012  012714  030125          MOV     #30125,@ADCR4   ;START CONVERSION
006016  105715                  TSTB    @ADCSR5         ;WAIT FOR DONE (SET VIA W.C.)
006020  100376                  BPL     .-2
006022  022714  030000          CMP     #30000,@ADCR4   ;CMP 1ST STATUS WRD TO 'CR'
006026  001401                  BEQ     .+4
006030  104400                  ERROR                   ;RANDOM DIDN'T PICK UP CORRECT STATUS WRD

                        ;*************************************************************************
                        ;TEST THAT THE A/D REG'S. ARE RESET ON W.C. OVRFLO USING 'RANDOM' MODE OPERATION
                        ;NOTE: THIS IS THE FIRST TIME MORE THAN 1 CONVERSION IS TAKEN UNDER RANDOM
                        ;*************************************************************************

006032  104002          TST76:  SCOPE
006034  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
006036  012701  030000          MOV     #30000,R1       ;SELECT; RAN,DMA,CH. '0'
006042  012702  014070          MOV     #RANBUF,R2      ;SETUP TO LOAD STATUS WORD BUFFER
006046  010122          TAGAN:  MOV     R1,(R2)+        ;LOAD CH.S '0-310'
006050  005201                  INC     R1
006052  022701  030311          CMP     #30311,R1       ;LOADED ALL CH.S?
006056  001373                  BNE     TAGAN           ;BRANCH IF NO
006060  012777  177500  173224  MOV     #-300,@ADWCR    ;LOAD W.C FOR 300 CONVERSIONS
006066  004767  002664          JSR     PC,SETREG       ;LOAD A/D REGS.
006072  012714  030125          MOV     #30125,@ADCR4   ;START CONVERSION
006076  105715                  TSTB    @ADCSR5         ;WAIT FOR DONE (SET VIA W.C.)
006100  100376                  BPL     .-2
006102  022777  014676  173204  CMP     #RANBUF+606,@ADSWR ;CHECK THAT STATUS WORD WAS UPDATED
006110  001401                  BEQ     .+4
006112  104400                  ERROR                   ;STATUS WRD BUFFER SHOULD = RANBUFF+606

006114  022777  015072  173176  CMP     #ADBUFF,@ADWRA  ;TEST THAT WRD REG. 'A' WAS RESET
006122  001401                  BEQ     .+4
006124  104400                  ERROR                   ;WORD REG.'A' WASN'T RESET VIA OVRFLO

006126  022777  015672  173166  CMP     #ADBUFF+600,@ADWRB      ;TEST THAT WRD REG 'B' WAS RESET
006134  001401                  BEQ     .+4
006136  104400                  ERROR                   ;WORD REG.'B' WASN'T RESET VIA OVRFLO

006140  022714  030277          CMP     #30277,@ADCR4   ;SHOULD = LAST CH.

006144  001401                  BEQ     .+4
006146  104400                  ERROR                   ;INITIAL CH. REG DOESN'T = LAST CH. CONVERTED
```

```
                                ;*************************************************************************
                                ;TEST THAT ONLY '8' LOCATIONS ARE MODIFIED IN RANDOM WITH W.C. @-10
                                ;*************************************************************************
006150  104002                TST77:  SCOPE
006152  104011                        INITAD                          ;CALL ROUTINE TO INITIALIZE THE 'A/D'
006154  012701  030000                MOV     #30000,R1               ;SELECT ;RAN,DMA,CH.'0'
006160  012702  014070                MOV     #RANBUF,R2              ;SETUP TO LOAD STATUS WORD BUFFER
006164  010122                TAGYA:  MOV     R1,(R2)+                ;LOAD CH.S'0-15'
006166  005201                        INC     R1
006170  022701  030015                CMP     #30015,R1               ;LOADED ALL CH.S?
006174  001373                        BNE     TAGYA
006176  012700  177763                MOV     #-15,R0
006202  012701  015072                MOV     #ADBUFF,R1              ;PRELOAD DATA AREA
006206  012721  125252        TAGYB:  MOV     #125252,(R1)+
006212  005200                        INC     R0
006214  001374                        BNE     TAGYB
006216  012777  177770  173[illegible]6    MOV     #-10,@ADWCR             ;LOAD W.C. FOR '8' CONVERSIONS
006224  004767  002526                JSR     PC,SETREG               ;LOAD A/D REGS.
006230  012714  030125                MOV     #30125,@ADCR4           ;START CONVERSIONS
006234  105715                        TSTB    @ADCSR5                 ;WAIT FOR DONE (SET VIA W.C.)
006236  100376                        BPL     .-2
006240  022767  125252  006644        CMP     #125252,ADBUFF+20 ;WERE ONLY '8' LOCATIONS MODIFIED?
006246  001401                        BEQ     .+4
006250  104400                        ERROR                           ;A/D TOOK MORE THAN '8' CONVERSIONS W/WC

                                ;*************************************************************************
                                ;TEST THAT BIT 09 SET IN A STATUS WORD WILL RESET THE SWAR
                                ;*************************************************************************
006252  104002                TST100: SCOPE
006254  104011                        INITAD          ;CALL ROUTINE TO INITIALIZE THE AD
006256  012701  030000                MOV     #30000,R1               ;SEL,RAM,DMACHO
006262  012702  014070                MOV     #RANBUF,R2              ;SET UP TO LOAD STATUS AND BUFF
006266  010122                TAGYC:  MOV     R1,(R2)+                ;LOAD CHS 0-10
006270  005201                        INC     R1
006272  022701  030010                CMP     #30010,R1               ;LOADED 10 CHANNELS
006276  001373                        BNE     TAGYC                   ;NOP NOT YET
006300  012701  014100                MOV     #RANBUF+10,R1           ;
006304  052711  001000                BIS     #1000,@R1               ;SET BIT 09 IN FINAL STATUS WORD
006310  012777  177767  172774        MOV     #-11,@ADWCR             ;LOAD WORD COUNT FOR 7 CONV.
006316  004767  002434                JSR     PC,SETREG               ;LOADA/D REGS
006322  012714  030125                MOV     #30125,@ADCR4           ;START CONV
006326  105715                        TSTB    @ADCSR5                 ;WAIT FOR DONE
006330  100376                        BPL     .-2
006332  022777  014074  172754        CMP     #RANBUF+4,@ADSWR        ;WAS STATUS WORD ADDRESS REG RESET
006340  001401                        BEQ     .+4
006342  104400                        ERROR                           ;SWAR DIDNT RESET WITH BIT 09 SET
```

```
                        ;*********************************************************************
                        ;THIS TEST WILL CHECK THE DOUBLE BUFFER FEATURES OF
                        ;THE RANDOM MODE BY PRELOADING THE DATA STORAGE
                        ;AREA WITH A KNOWN VALUE. A WINDOW OF 32 WORDS IS
                        ;PLACED BETWEEN THE BUFFERS, THIS WINDOW IS TESTED
                        ;TO VERIFY THAT NO DATA HAS BEEN STORED IN THIS AREA.
                        ;ON THE FIRST WORD COUNT OVERFLOW THE "GO" BIT IS
                        ;SET AND THE "XFER CHEK" BIT IS CLEAR ENABLING BUFFER
                        ;B TO BE FILLED. THE "XFER CHEK" BIT IS THEN SET PREVENTING
                        ;THE WORD COUNT OVERFLOW FROM BUFFER B TO START FILLING
                        ;BUFFER A AGAIN.
                        ;WHILE BUFFER B IS BEING FILLED THE CONTENTS OF
                        ;BUFFER A IS TESTED FOR BEING MODIFIED.
                        ;BUFFER A IS THEN REFILLED WITH A KNOWN VALUE, THEN THE
                        ;WORD COUNT OVERFLOW FROM BUFFER B IS WAITED FOR. WHEN THE
                        ;WORD COUNT IS DETECTED A STALL LOOP IS RUN. THEN BUFFER B
                        ;IS CHECKED TO VERIFY THAT THE "GO" BIT PREVENTED
                        ;THE OVERFLOW FROM BUFFER A FROM LOADING DATA INTO BUFFER B.
                        ;*********************************************************************

006344  104002          TST101: SCOPE
006346  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE A/D
006350  104012                  SAVREG                  ;SAVE ALL REG. ON STACK
006352  012700  177500          MOV     #-300,R0        ;PRE LOAD COUNTER
006356  012701  015072          MOV     #ADBUFF,R1      ;GET DATA BUFFER TABLE
006362  012721  152525  1$:     MOV     #152525,(R1)+   ;LOAD BUFFER AREA
006366  005200                  INC     R0
006370  001374                  BNE     1$
006372  012777  177700  172712  MOV     #-100,@ADWCR    ;SET W.C. FOR 100 OCTAL
006400  004767  002352          JSR     PC,SETREG       ;LOAD A/D REGISTERS
006404  062777  000100  172710  ADD     #100,@ADWRB     ;OFFSET REGISTER B FROM A BY 100 OCTAL
006412  012701  014070          MOV     #RANBUF,R1      ;GET STATUS WORD TABLE POINTER
006416  012700  177700          MOV     #-100,R0        ;PRE LOAD COUNTER FOR 100 STATUS WORDS.
006422  005003          2$:     CLR     R3              ;START EACH TIME WITH CH 0
006424  010304          3$:     MOV     R3,R4           ;PUT
006426  062704  010000          ADD     #10000,R4       ;RANDOM.DMA (BASE VALUE OF STATUS)
006432  010421                  MOV     R4,(R1)+
006434  005200                  INC     R0              ;
006436  001405                  BEQ     4$              ;
006440  005203                  INC     R3              ;THE RANDOM CHANNELPARE ONLY 0,1,2,3
006442  022703  000004          CMP     #4,R3           ;IF NOW A 4 WE'RE TOO HIGH
006446  001366                  BNE     3$              ;RETURN AND LOAD NEXT CH.
006450  000764                  BR      2$              ;RETURN AND START FROM CH0.
006452  052741  001000  4$:     BIS     #1000,-(R1)     ;SET FINAL CHANNEL FLAG BIT 9
006456  104013                  GETREG                  ;RESTORE ALL REGS.
006460  004767  003134          JSR     PC,LDINTR       ;LOAD THE INT. VECTOR
006464  006500                  TAGRAN                  ;INT RETURN TAG
006466  012714  010000          MOV     #10000,@ADCR4   ;START THE CONV. BY LOADING THE CR.
006472  012715  000101          MOV     #101,@ADCSR5    ;SET THE INT. ENABLE AND GO BIT
006476  000001                  WAIT                    ;WAIT FOR INT. FROM A/D

                        ;RETURN HERE FROM 1ST INTERRUPT AND TEST THAT BUFFER A HAS BEEN MODIFIED

006500  022626          TAGRAN: POP2SP                  ;RESET THE STACK
006502  012715  000001          MOV     #1,@ADCSR5      ;INC GO BIT
006506  012701  015072          MOV     #ADBUFF,R1      ;POINT TO TABLE A
```

```
006512  012700  177700          MOV     #-100,R0        ;LOAD COUNTER TO VERIFY TABLE A
006516  022721  152525  1$:     CMP     #152525,(R1)+   ;TEST THAT DATA IN TABLE A HAS BEEN MOD.
006522  001403                  BEQ     2$              ;IF EQUAL REPORT AN ERROR
006524  005200                  INC     R0              ;AT END OF TABLE?
006526  001373                  BNE     1$              ;NOP TRY AGAIN
006530  000401                  BR      3$              ;YES!
006532  104400          2$:     ERROR                   ;REPORT TABLE A NOT MODIFIED IN RANDOM
                        ;NOW SET UP INTERRUPT FROM WORD COUNT OVERFLOW OF TABLE B

006534  012777  006562  172564  3$:     MOV     #TAGRAB,@ADINT  ;LOAD THE INTERRUPT VECTOR
006542  012777  000340  172560          MOV     #340,@ADLVL             ;INT. RETURN TAG
006550  052715  000100                  BIS     #100,@ADCSR5            ;ENABLE THE INTERRUPT
006554  005077  172510                  CLR     @PSW
006560  000001                  WAIT                    ;WAIT FOR INTERRUPT

                        ;ENTER HERE FROM INTERRUPT FROM TABLE B
                        ;TABLE A IS NOW BEING FILLED AGAIN


006562  022626          TAGRAB: POP2SP                  ;RESET THE STACK

                        ;NOW VERIFY THAT THE WINDOW GAP HAS NOT BEEN WRITTEN OVER

006564  042715  000100          BIC     #100,@ADCSR5            ;CLR INT. ENABLE
006570  012701  015272          MOV     #ADBUFF+200,R1  ;LOAD POINTER TO WINDOW AREA
006574  012700  177740          MOV     #-40,R0         ;PRESET COUNTER
006600  022721  152525  1$:     CMP     #152525,(R1)+   ;CHECK DATA IN WINDOW AREA
006604  001003                  BNE     2$              ;REPORT ERROR DATA IS WRITTEN IN WINDOW AREA
006606  005200                  INC     R0              ;UP COUNTER
006610  001373                  BNE     1$              ;GO BACK IF NOT DONE YET
006612  000401                  BR      3$
006614  104400          2$:     ERROR                   ;DATA WAS WRITTEN IN WINDOW AREA

                        ;REFILL TABLE B WITH KNOWN DATA
006616  012701  015372  3$:     MOV     #ADBUFF+300,R1  ;POINT AT TABLE B
006622  012700  177700          MOV     #-100,R0        ;PRELOAD COUNTER
006626  012721  152525  4$:     MOV     #152525,(R1)+   ;LOAD TABLE B
006632  005200                  INC     R0
006634  001374                  BNE     4$
006636  012777  006664  172462          MOV     #SYNRET,@ADINT          ;LOAD INT VECTOR
006644  012777  000340  172456          MOV     #340,@ADLVL             ;LOAD THE NEW PROCESSOR PRI.
006652  052715  000100                  BIS     #100,@ADCSR5            ;ENABLE THE INTERRUPT
006656  005077  172406                  CLR     @PSW                    ;CLEAR PROCESSOR STATUS
006662  000001                  WAIT                    ;NOW WAIT FOR INT. FROM TABLE A

                        ;TEST THAT THE A/D STOPPED BY TESTING THAT NO DATA WAS WRITTEN  INTO BUFFER B

006664  022626          SYNRET: POP2SP          ;RESTORE STACK
006666  042715  000100          BIC     #100,@ADCSR5            ;DISABLE THE INTERRUPT
006672  012700  177700          MOV     #-100,R0
006676  005200                  INC     R0
006700  001376                  BNE     .-2
006702  012701  015372          MOV     #ADBUFF+300,R1  ;LETS LOOK AT FIRST WORD IN TABLE B
006706  022711  152525          CMP     #152525,(R1)
006712  001401                  BEQ     6$
006714  104400                  ERROR                   ;A/D FAILED TO STOP WITH GO=0
```

```
        006716  000240                  6$:     NOP
```

```
                        ;*****************************************************************************
                        ;TEST 'INCREMENT MEMORY' MODE
                        ;THIS ROUTINE EXAMINES THE LOCATION 'INCFLG' (A SOFTWARE FLAG INDICATING
                        ;SUFFICIENT CORE). IF CLEARED, THIS TEST IS SKIPPED
                        ;INCREMENT MEMORY WILL BE TESTED IF THE PDP11 HAS AT LEAST
                        ;8K OF CORE .THE (IOR)IS PRESET WITH 034000
                        ;WHICH WILL PLACE THE INCREMENTED LOCATION AT HIGH END OF
                        ;8K. THE (SWAR) IS READ AFTER TAKING ONE INCREMENT MEMORY CONVERT
                        ;TO DETERMINE THE INCREMENTED LOCATION. THE LOCATION IS THEM
                        ;TESTED FOR HAVING BEEN MODIFIED.

                        ;*****************************************************************************

006720  104002          TST102: SCOPE
006722  005767  005072          TST     INCFLG          ;TEST IF CORE IS AVAILABLE
006726  001431                  BEQ     TAGAM           ;EXIT IF NOT
006730  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
006732  012701  020000          MOV     #20000,R1       ;CLEAR THE LOAD ZONE
006736  005021          CLRCOR: CLR     (R1)+           ;CLR CORE
006740  022701  040000          CMP     #40000,R1       ;DONE
006744  001374                  BNE     CLRCOR          ;BRANCH IF NOT
006746  012777  034000  172350  MOV     #34000,@ADADR   ;LOAD OFFSET IN IOR
006754  012777  177777  172330  MOV     #-1,@ADWCR      ;LOAD W.C FOR 1 CONVERSION
006762  004767  001770          JSR     PC,SETREG       ;SET UP A/D REG'S.
006766  012714  001000          MOV     #1000,@ADCR4    ;LOAD FINAL CH.
006772  012714  112000          MOV     #112000,@ADCR4  ;SELECT:SEQ, DMA, INC.MEM & START
006776  104007                  CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
007000  104400                  ERROR                   ;DONE FAILED TO SET
007002  017701  172306          MOV     @ADSWR,R1       ;OBTAIN ADDRESS OF DATA FROM STATUS
007006  022711  000001          CMP     #1,(R1)         ;WAS LOCATION INCREMENTED?
007012  001401          TAGAM:  BEQ     .+4             ;BRANCH IF YES
007014  104400                  ERROR                   ;SELECTED MEMORY ADDR. WASN'T INCREMENTED
```

```
                        ;*****************************************************************************
                        ;TEST INCREMENT MEMORY 'DATA OVRFLO'
                        ;THIS ROUTINE EXAMINES THE LOCATION 'INCFLG' (A SOFTWARE FLAG INDICATING
                        ;SUFFICIENT CORE).  IF CLEARED, THIS TEST IS SKIPPED OTHERWISE 'DATA
                        ;OVERFLOW' IS TESTED.  THIS IS DONE VIA LOADING ALL POSSIBLE ADDRESSES
                        ;TO BE INCREMENTED WITH '-1' AND TAKING ONE 'DMA' CONVERTION AT A TIME.
                        ;A TEST IS THEN MADE TO SEE IF 'DATA OVRFLO' WAS SET.
                        ;*****************************************************************************

007016  104002          TST103: SCOPE
007020  005767  004774          TST     INCFLG          ;TEST IF CORE IS AVAILABLE
007024  001435                  BEQ     TAGYF           ;EXIT IF NOT
007026  104011                  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
007030  012777  020000  172266  MOV     #20000,@ADADR   ;LOAD 'A/D OFFSET' REG.
007036  012701  020000          MOV     #20000,R1       ;START AT BOTTOM OF 8K
007042  012721  177777  LODCOR: MOV     #-1,(R1)+       ;LOAD MEM. WITH '-1'
007046  020127  040000          CMP     R1,#40000       ;DONE
007052  001373                  BNE     LODCOR          ;BRANCH IF NOT
007054  012777  177777  172230  MOV     #-1,@ADWCR      ;LOAD W.C FOR 1 CONVERSION
007062  004767  001670          JSR     PC,SETREG       ;SET UP A/D REG'S.
007066  012714  112000          MOV     #112000,@ADCR4  ;SELECT: SEQ,DMA,INC.MEM & START
007072  104007                  CKDONE                  ;SUBROUTINE TO CHECK FOR 'A/D DONE'
007074  104400                  ERROR                   ;DONE FAILED TO SET
007076  017701  172212          MOV     @ADSWR,R1       ;OBTAIN ADDRESS FROM STATUS REG.
007102  005711                  TST     (R1)            ;CHECK IF ADDRESS WAS INCREMENTED TO '0'
007104  001401                  BEQ     .+4
007106  104400                  ERROR                   ;SELECTED ADDR WASN'T CLEARED
007110  032715  010000          BIT     #10000,@ADCSR5  ;TEST IF OVRFLO SET
007114  001001                  BNE     .+4
007116  104400                  ERROR                   ;DATA OVRFLO FAILED TO SET
007120  000240          TAGYF:  NOP                     ;DATA OVRFLO FAILED TO SET
```

```
                                        ;TEST THAT DATA CAN BE STORED IN THE HIGHEST 1K OF MEMORY USING SEQ. MODE

007122  104002                  TST104: SCOPE
007124  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
007126  012777  177300  172156          MOV     #-500,@ADWCR    ;SET UP FOR '640' CONVERSIONS
007134  016777  004664  172156          MOV     CORMAX,@ADWRA
007142  016777  004656  172152          MOV     CORMAX,@ADWRB
007150  062777  001200  172144          ADD     #1200,@ADWRB    ;OFFSET WRD REG.'B' VIA '1200' WRDS
007156  016700  004642                  MOV     CORMAX,R0
007162  012701  001000                  MOV     #1000,R1
007166  012720  152525                  MOV     #152525,(R0)+   ;PRE-LOAD DATA AREA
007172  005301                          DEC     R1
007174  001374                          BNE     .-6
007176  004767  002416                  JSR     PC,LDINTR       ;SET UP INTERRUPT ADDRESS
007202  007220                          TAGA0
007204  012714  111007                  MOV     #111007,@ADCR4  ;SELECT; DMA, FINAL CH. '7'
007210  005215                          INC     @ADCSR5         ;SET 'GO'
007212  012714  130007                  MOV     #130007,@ADCR4  ;SELECT;SEQ.,DMA,SINGLE CH.
007216  000001                          WAIT                    ;WAIT FOR INTERRUPT

007220  052715  004000          TAGA0:  BIS     #4000,@ADCSR5   ;CLR ALL FLAGS+'GO BIT'
007224  022626                          POP2SP                  ;RESET STACK POINTER
007226  004767  002366                  JSR     PC,LDINTR       ;RE-ENABLE INTERRUPT
007232  007236                          TAGAAQ
007234  000001                          WAIT                    ;WAIT FOR 2ND INTERRUPT

007236  052715  004000          TAGAAQ: BIS     #4000,@ADCSR5   ;CLR ALL FLAGS
007242  022626                          POP2SP                  ;RESET STACK POINTER
007244  004767  002404                  JSR     PC,CLRINT       ;CLR INTERRUPT
007250  016700  004550                  MOV     CORMAX,R0
007254  012701  001000                  MOV     #1000,R1
007260  022720  152525          TAGAAR: CMP     #152525,(R0)+   ;SEE IF MEMORY WAS MODIFIED
007264  001002                          BNE     .+6
007266  104400                          ERROR                   ;MEMORY WASN'T MODIFIED ON HIGH MEM. TRANS.
007270  000402                          BR      .+6             ;EXIT ON ERROR
007272  005301                          DEC     R1
007274  001371                          BNE     TAGAAR
007276  000240          TAGAP:  NOP                             ;EXIT ADDRESS IF RUN VIA ACT11
```

```
                                        ;TEST THAT THE 'A/D' WILL OPERATE IN THE HIGHEST 1K OF MEMORY IN RANDOM MODE

007300  104002                  TST105: SCOPE
007302  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
007304  012777  177500  172000          MOV     #-300,@ADWCR    ;SET UP FOR '192' CONVERSIONS
007312  016777  004506  171774          MOV     CORMAX,@ADSWR   ;LOAD STATUS WORD REG.
007320  016777  004500  171772          MOV     CORMAX,@ADWRA
007326  062777  001000  171764          ADD     #1000,@ADWRA    ;OFFSET DATA BUFFER '500' WRDS FROM STATUS BUFFER
007334  012777  000600  171760          MOV     #600,@ADWRB     ;LOAD REG WITH DOUBLE THE W.C.
007342  067777  171752  171752          ADD     @ADWRA,@ADWRB   ;OFFSET 'B' DOUBLE THE W.C. FROM 'A'
007350  017700  171740                  MOV     @ADSWR,R0
007354  012701  000310                  MOV     #310,R1         ;LOAD '310' STATUS WORDS
007360  012720  030125                  MOV     #30125,(R0)+
007364  005301                          DEC     R1
007366  001374                          BNE     .-6
007370  017700  171724                  MOV     @ADWRA,R0       ;PRE-LOAD DATA AREA.
007374  012701  000600                  MOV     #600,R1
007400  012720  152525                  MOV     #152525,(R0)+
007404  005301                          DEC     R1
007406  001374                          BNE     .-6
007410  004767  002204                  JSR     PC,LDINTR       ;SET UP INTERRUPT ADDRESS
007414  007424                          TAGAQ
007416  012714  030007                  MOV     #30007,@ADCR4   ;SELECT; RANDOM, DMA
007422  000001                          WAIT

007424  052715  004000          TAGAQ:  BIS     #4000,@ADCSR5   ;CLR ALL FLAGS
007430  022626                          POP2SP                  ;RESET STACK POINTER
007432  004767  002216                  JSR     PC,CLRINT       ;CLR INTERRUPT
007436  017700  171656                  MOV     @ADWRA,R0
007442  012701  000300                  MOV     #300,R1
007446  022720  152525          TAGAR:  CMP     #152525,(R0)+   ;SEE IF MEMORY WAS MODIFIED
007452  001002                          BNE     .+6
007454  104400                          ERROR                   ;LAST '1K' OF MEM. WASN'T MODIFIED
007456  000402                          BR      .+6             ;EXIT ON ERROR
007460  005301                          DEC     R1
007462  001371                          BNE     TAGAR

                                        ;EXAMINE THAT 'WORD REG.B' WASN'T MODIFIED

007464  016700  004334                  MOV     CORMAX,R0       ;LOAD 'B' STARTING ADDR.
007470  062700  001600                  ADD     #1600,R0        ;SET UP ADDRESS OF BUFFER 'B'
007474  012701  000300                  MOV     #300,R1
007500  022720  152525          TAGAS:  CMP     #152525,(R0)+
007504  001401                          BEQ     .+4
007506  104400                          ERROR                   ;BUFFER 'B' WAS MODIFIED WITH 'GO' CLEARED.
007510  005301                          DEC     R1
007512  001372                          BNE     TAGAS
```

```
                        ;*************************************************************************
                        ;LAST BUT IN NO WAY LEAST IS THE ' B U R S T  M O D E '  TEST.
                        ;THIS TEST PRE-LOADS THE DATA BUFFER WITH KNOWN DATA THAN SETS UP TO
                        ;TAKE '500' SEQUENTIAL CONVERSIONS USING THE BURST MODE FEATURE.
                        ;*************************************************************************

007514  104002          TST106: SCOPE
007516  104011                  INITAD                      ;CALL ROUTINE TO INITIALIZE THE 'A/D'
007520  012701  015072          MOV     #ADBUFF,R1          ;SET UP TO PRE-LOAD DATA BUFFER
007524  012721  125252          MOV     #125252,(R1)+
007530  022701  016056          CMP     #ADBUFF+500.,R1     ;DONE?
007534  001373                  BNE     .-10                ;NO
007536  012777  177014  171546  MOV     #-500.,@ADWCR       ;SET UP TO TAKE '500' CONVERSIONS
007544  004767  001206          JSR     PC,SETREG           ;SET UP THE A/D REG.'S
007550  004767  002044          JSR     PC,LDINTR           ;LOAD THE A/D INTERRUPT ADDRESS
007554  007574                  TAG1B                       ;TO INTERRUPT TO HERE
007556  012714  001777          MOV     #1777,@ADCR4        ;LOAD FINAL CH.
007562  012715  000102          MOV     #102,@ADCSR5        ;SET 'BURST' ENABLE BIT & INTR.ENB.
007566  012714  110000          MOV     #110000,@ADCR4      ;SELECT: SEQ.,DMA, INITIAL CH. '0'
007572  000001                  WAIT                        ;NOW HOPE FOR THE BEST

                        ;ENTER HERE ON THE A/D INTERRUPT

007574  052715  004000  TAG1B:  BIS     #4000,@ADCSR5       ;CLR ALL FLAGS
007600  022626                  POP2SP                      ;RESET STACK
007602  004767  002046          JSR     PC,CLRINT           ;CLR OUT INTERRUPT ADDRESS
007606  012701  015072          MOV     #ADBUFF,R1          ;SET UP TO EXAMINE THE BUFFER
007612  022721  125252  TAG1C:  CMP     #125252,(R1)+       ;WAS LOCATION MODIFIED?
007616  001002                  BNE     .+6                 ;YES
007620  104400                  ERROR                       ;DATA BUFFER WASN'T MODIFIED?
007622  000403                  BR      .+10                ;EXIT ON ERROR
007624  022701  015572          CMP     #ADBUFF+500,R1      ;EXAMINED ENTIRE BUFFER?
007630  001370                  BNE     TAG1C               ;NO

                        ;*************************************************************************
                        ;TEST COMPLETE
                        ;*************************************************************************

007632  104002          TST107: SCOPE

007634  104000                  PRINT                       ;TYPE 'LOGIC' TO INDICATE PASS COMPLETION
007636  013340                  MES6
007640  000167  172136          JMP     RESTRT              ;RESTART TEST
```

```
                                ;*****************************************************************************
                                ;DATA UPDATE TEST
                                ;*****************************************************************************

                                ;THE DATA 'UPDATE' TEST IS AN OPERATOR INTERVENTION TEST USED TO TEST
                                ;THE A/D 'DATA BUFFER' & UPDATE LOGIC. THE TEST REQUESTS '4' SPECIFIC
                                ;VOLTAGES (2 POS. & 2 NEG) TO SUPPLIED TO CH.'0'. A CONVERSION IS THEN
                                ;TAKEN AT EACH OF THESE VOLTAGES AND THEN THE '4' BUFFERED DATA WORDS ARE
                                ;READ. THE PROGRAM CHECKS THE SIGN OF THE READ DATA TO DETERMINE IF THE
                                ;DATA WAS BUFFERED CORRECTLY.

007644  004767  172056          DATA:   JSR     PC,SETUP        ;INITIALIZE TEST
007650  012767  007662  004126          MOV     #DATA1,AVECTR   ;LOAD THE '1A' VECTOR ADDRESS
007656  104000                          PRINT
007660  013361                          MES7                    ;TEXT 'DATA UPDATE TEST'
007662  104011                  DATA1:  INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
007664  104000                          PRINT
007666  013311                          CRLF
007670  012700  000002                  MOV     #2,R0
007674  104000                  DATA2:  PRINT
007676  013510                          MES8                    ;TEXT 'SUPPLY +5V TO CH. '0'
007700  004767  000062                  JSR     PC,WATINP       ;WAIT FOR INPUT & ST. CNVRT
007704  104000                          PRINT
007706  013530                          MES9                    ;TEXT 'SUPPLY -5V TO CH. '0'
007710  004767  000052                  JSR     PC,WATINP       ;WAIT FOR INPUT & ST. CNVRT
007714  005300                          DEC     R0
007716  001366                          BNE     DATA2
007720  012700  000002                  MOV     #2,R0
007724  005001                          CLR     R1
007726  011367  004114          DATA3:  MOV     @ADDBR3,KSTOR4  ;READ & STORE DATA
007732  100001                          BPL     .+4             ;BRANCH IF POS.
007734  174400                          ERROR                   ;1ST OR 3RD READ DATA WASN'T POS.
007736  005201                          INC     R1
007740  011367  004102                  MOV     @ADDBR3,KSTOR4  ;READ & STORE DATA
007744  100401                          BMI     .+4             ;BRANCH IF MINUS
007746  104400                          ERROR                   ;2ND OR 4TH READ DATA WASN'T NEG.
007750  005201                          INC     R1
007752  005300                          DEC     R0
007754  001364                          BNE     DATA3
007756  105777  171314                  TSTB    @TPS
007762  100375                          BPL     .-4
007764  000736                          BR      DATA1

007766  104005                  WATINP: TTYIN                   ;WAIT FOR 'CR'
007770  005014                          CLR     @ADCR4          ;START CONVERSION
007772  105715                          TSTB    @ADCSR5         ;WAIT FOR DONE
007774  100376                          BPL     .-2
007776  052715  004000                  BIS     #4000,@ADCSR5   ;CLR ALL A/D FLAGS
010002  000207                          RTS     PC
```

```
;*************************************************************************
;THE CONVERTION SYNC TEST WILL TAKE CONVERTIONS UTILIZING THREE
;MODES OF CONVERTION SYNC.
;
;                1. EXTERNAL
;                2. INTERNAL
;                3. INTERNAL AND EXTERNAL (ALTERNATE) RANDOM MODE ONLY
;
;THE 1ST TEST WILL TAKE ONE CONVERTION IN THE SINGLE CHANNEL
;MODE REQUIREING ONE EXTERNAL TRIGGER PULSE.
;
;THE 2ND TEST IS RUN IN SEQUENTIAL CHANNEL - PMA MODE USING THE
;EXTERNAL SYNC (BIT 11 OF THE CONTROL REGISTER). ONE EXTERNAL PULSE SHOULD
;INITIATE THE ENTIRITY OF WORD COUNT.
;
;THE 3RD TEST WILL TAKE CONVERTIONS IN THE RANDOM CHANNEL - DMA MODE
;WITH ALL CONVERTIONS BEING INITIATED INDIVIDUALLY BY AN EXTERNAL
;CLOCK PULSE
;
;THE 4TH TEST WILL TAKE CONVERTIONS IN THE RANDOM CHANNEL - DMA
;MODE TAKING CONVERTIONS ALTERNATELY BY INTERNAL THEN EXTERNAL
;SYNC.
;
;THIS TEST REQUIRES THAT AN EXTERNAL PULSE GENERATOR (OR EQUIVELENT)
;BE CONNECTED TO THE EXTERNAL TRIGGER INPUT OF THE MSO1 MODULE
;THE REPTITION RATE OF THE EXTERNAL TRIGGER PULSES SHOULD BE LESS
; THAN OR EQUAL TO 666.6 HERTZ.
;
;IF THE UPPLIED EXTERNAL TRIGGER PULSE RATE ISGREATER THAN
;THE TIME REQUIRED TO TAKE 100 (DEC.) CONVERTIONS THE CONVERTION IN PROGRESS
;ERROR WILL INTERRUPT THE TEST AND REPORT ON THE CONSOLE DEVICE
;IF THE EXTERNAL TRIGGER PULSES FAIL TO INITIATE THE CONVERTIONS
;SOFTWARE LOOPS WILL TIME OUT AND REPORT THE FAILURE ON THE
;CONSOLE DEVICE
;
;
;THIS TEST IS ENTERED FROM THE MONITOR BY TYPING AN "S" FOLLOWED
;BY A C.R.
;
;THE TEST WILL RESPOND BY PRINTING THE INTRODUCTORY MESSAGE
;FOLLOWED BY INSTRUCTIONS TO TYPE A CARRIAGE RETURN TO START THE TEST
;
;*************************************************************************
;TAKE A CONVERTION IN THE SEQUENTIAL CHANNEL MODE INITIATED BY
;AN EXTERNAL TRIGGER PULSE
;*************************************************************************
010004  016706  003772          SYNCO:  MOV     STACK,SP                ;INIT THE STACK
010010  104000  013742                  PRINT,MES43                     ;CONVERTION SYNC TEST
010014  104000                          PRINT
010016  013456                          MES44                   ;PRINT MESSAGE TO STRIKE A C.R.
010020  104005                          TTYIN                   ;START TEST WITH A C.R.
010022  104000  013311                  PRINT,CRLF              ;OUTPUT A C.R.L.F.
010026  000005                          RESET
010030  004767  171672                  JSR     PC,SETUP        ;INIT REGISTERS AND SOFT FLAGS
010034  012767  010042  002646  CYC:    MOV     #SYNC,RETURN
```

```
010042  104002                  SYNC:   SCOPE
010044  104011                          INITAD                  ;INITIALIZE A/D
010046  012777  177777  171236          MOV     #-1,@ADWCR      ;SET WORD COUNT FOR 1
010054  004767  000676                  JSR     PC,SETREG       ;LOAD REGISTERS OF THE A/D
010060  004767  001534                  JSR     PC,LDINTR
010064  010114                          SYNC1
010066  012714  115000                  MOV     #115000,@ADCR4  ;SEQ, DMA, EXTERNAL SYNC FINAL CH. + CH 0.
010072  012714  114000                  MOV     #114000,@ADCR4  ;SEQ, DMA, EXTERNAL SYNC + CH 0
010076  005000                          CLR     R0              ;LOAD REG 0 FOR WAIT LOOP
010100  005200                  1$:     INC     R0              ;WAIT LONG DURATION FOR CONVERTION TO BE DONE
010102  001376                          BNE     1$
010104  042715  000100                  BIC     #100,@ADCSR5
010110  104400                          ERROR                   ;EXTERNAL TRIGGER DIDN'T START A/D CONV.
010112  000401                          BR      SYNC1+2
                        ;*****************************************************************************
                        ;TAKE (100 DEC.) CONVERTIONS IN THE SINGLE CHANNEL MODE INITIATED BY
                        ;AN EXTERNAL TRIGGER PULSE.
                        ;*****************************************************************************
010114  022626          SYNC1:  POP2SP          ;RESET THE STACK
010116  005715                  TST     @ADCSR5         ;WAS INTERRUPT FROM ERROR?
010120  100001                  BPL     .+4
010122  104400                  ERROR                   ;STATUS ERROR FROM ADF CSR
010124  104002                  SCOPE
010126  104011                  INITAD                  ;INITIALIZE THE A/D
010130  012701  015072          MOV     #ADBUFF,R1      ;PRELOAD DATA AREA
010134  012721  152525          MOV     #152525,(R1)+   ;LOAD CONSTANT IN DATA AREA
010140  022701  015274          CMP     #ADBUFF+202,R1  ;LOADED 100 LOCATIONS YET
010144  001373                  BNE     1$              ;IF NOT DO-IT
010146  004767  001446          JSR     PC,LDINTR       ;LOAD INT VECTOR
010152  010220                  SYNRE1                  ;  THIS IS THE RETURNING ADDRESS
010154  012777  177634  171130  MOV     #-144,@ADWCR    ;YES 100 DONE NOW SET THE WORD COUNT
010162  004267  000570          JSR     R2,SETREG       ;LOAD A/D REGISTERS
010166  012700  177634          MOV     #-144,R0        ;# OF CONVERTIONS.
010172  012714  115000          MOV     #115000,@ADCR4  ;SEQ, DMA, EXTERNAL, FINAL + CH 0
010176  012714  114000          MOV     #114000,@ADCR4  ;SEQ, DMA, EXTERNAL, + CH 0 (START CONV.)
010202  005001          2$:     CLR     R1              ;LOAD VALUE FOR EACH CONVERTION
010204  005201                  INC     R1              ;WAIT FOR INDIVIDUAL CONVERTION
010206  001376                  BNE     .-2             ;WAIT FOR INDIVIDUAL CONVERTION
010210  005200                  INC     R0              ;WAIT 100 TIMES
010212  001373                  BNE     2$              ;IF NOT 100 TIMES GO BACK
010214  104400                  ERROR                   ;DONE FAILED TO SET (100 CONVERTS) STARTED EXTERNAL.
010216  000417                  BR      SYNC2           ;GET TO SCOPE ROUTINE AFTER AN ERROR
010220  022626          SYNRE1: POP2SP                  ;RESTORE STACK
010222  005715                  TST     @ADCSR5         ;TEST FOR A STATUS ERROR
010224  100001                  BPL     .+4
010226  104400                  ERROR                   ;INTERRUPT WAS FROM ADF STATUS
010230  012701  015072          MOV     #ADBUFF,R1      ;VERIFY DATA WAS REALLY READ-IN
010234  012700  177634          MOV     #-144,R0        ;# OF CONVERTS TO BE VERIFIED
010240  022721  152525  4$:     CMP     #152525,(R1)+   ;VERIFY DATA WAS READ-IN
010244  001403                  BEQ     5$
010246  005200                  INC     R0              ;100 CONVERTIONS DONE YET?
010250  001373                  BNE     4$              ;NO
010252  000401                  BR      SYNC2           ;YES
010254  104400          5$:     ERROR                   ;THE DONE WAS SET BUT BUFFER WASN'T MODIFIED
```

```
                                ;*************************************************************************
                                ;TAKE 100 (OCTAL) CONVERTIONS IN THE RANDOM-DMA MODE ALL SYNCRONIZED
                                ;INDIVIDUALLY BY EXTERNAL TRIGGER PULSES.
                                ;*************************************************************************

010256  104002                  SYNC2:  SCOPE
010260  104011                          INITAD                          ;INITIALIZE THE ADF
010262  012701  015072                  MOV     #ADBUFF,R1              ;PRELOAD DATA AREA
010266  012721  152525          1$:     MOV     #152525,(R1)+           ;LOAD CONSTANT INTO DATA AREA
010272  022701  015274                  CMP     #ADBUFF+202,R1          ;IS AREA LOADED YET?
010276  001373                          BNE     1$
010300  012701  014070                  MOV     #RANBUF,R1              ;YES NOW LOAD STATUS WORDS
010304  012721  014000          2$:     MOV     #014000,(R1)+           ;RANDOM, DMA, EXT, CH=0
010310  022701  014272                  CMP     #RANBUF+202,R1          ;100 WORDS YET?
010314  001373                          BNE     2$
010316  004767  001276                  JSR     PC,LDINTR
010322  010364                          SYNRE2
010324  012777  177634  170760          MOV     #-144,@ADWCR            ;SET WORD COUNT FOR 100 CONVERTIONS
010332  004767  000420                  JSR     PC,SETREG               ;SET UP A/D REGISTERS
010336  012714  014000                  MOV     #014000,@ADCR4          ;LOAD CONTROL REGISTER RAN, DMA, EXT. TRIG.
010342  012701  177634                  MOV     #-144,R1                ;NUMBER OF CONVERTIONS WAITED FOR
010346  005000          3$:     CLR     R0                              ;DURATION OF EACH CONVERTION
010350  005200          4$:     INC     R0                              ;DO A SINGLE DURATION HERE
010352  001376                  BNE     4$                              ;LOOP TILL DONE
010354  005201                  INC     R1                              ;COUNT DURATIONS
010356  001373                  BNE     3$                              ;LOOP TILL 100 DURATIONS DONE
010360  104400                  ERROR                                   ;DONE FAILED TO SET ON 100 RANDOM, (EXT.TRIG) CONVERTS
010362  000416                  BR      SYNC3                           ;WITH ERROR
010364  022626          SYNRE2: POP2SP                                  ;RESTORE STACK
010366  005715                  TST     @ADCSR5                         ;TEST FOR ASTATUS ERROR
010370  100001                  BPL     .+4
010372  104400                  ERROR                                   ;INTERRUPT WAS FROM STATUS ERROR
010374  012701  015072          MOV     #ADBUFF,R1                      ;VERIFY THAT DATA WAS MODIFIED
010400  022721  152525  6$:     CMP     #152525,(R1)+                   ;
010404  001404                  BEQ     7$
010406  022701  015272          CMP     #ADBUFF+200,R1                  ;HAVE WE DONE 100 VERIFICATIONS YET?
010412  001372                  BNE     6$                              ;IF NOT LOOP AGAIN
010414  000401                  BR      SYNC3                           ;YES, NO ERROR
010416  104400          7$:     ERROR                                   ;FAILED TO MODIFY BUFFER AREA (DONE SET)
                                ;*************************************************************************
                                ;TAKE 100 (OCTAL CONVERTIONS IN THE RANDOM-DMA MODE.  THE CONVERTS ARE
                                ;SYNCRONIZED ALTERNATELY "INTERNAL" AND "EXTERNAL"
                                ;*************************************************************************

010420  104002                  SYNC3:  SCOPE
010422  104011                          INITAD                          ;INITIALIZE THE ADF
010424  012701  015072                  MOV     #ADBUFF,R1              ;GET POINTER TO DATA AREA
010430  012721  152525          1$:     MOV     #152525,(R1)+           ;LOAD DATA AREA
010434  022701  015274                  CMP     #ADBUFF+202,R1          ;DONE YET
010440  001373                          BNE     1$                      ;LOOP TILL DONE
010442  004767  001152                  JSR     PC,LDINTR
010446  010542                          SYNRE3
010450  012701  014070                  MOV     #RANBUF,R1              ;GET STATUS WORD TABLE POINTER
010454  012721  014000          2$:     MOV     #014000,(R1)+           ;RAN, DMA, EXT, TRIG AND CH=0
010460  012721  010000                  MOV     #010000,(R1)+           ;RAN, DMA, CH=0
```

```
010464  022701  014404                  CMP     #RANBUF+314,R1  ;LOAD 100 STATUS WORDS
010470  001371                          BNE     2$
010472  052711  001000                  BIS     #1000,(R1)      ;SET THE FINAL WORD STATUS BIT
010476  012777  177634  170606          MOV     #-144,@ADWCR    ;SET WORD COUNT FOR 100
010504  004767  000246                  JSR     PC,SETREG       ;SET UP A/D REG.
010510  012714  014000                  MOV     #014000,@ADCR4  ;LOAD CONTROL REG
010514  012701  177634                  MOV     #-144,R1        ;# OF CONVERTS
010520  005000                  3$:     CLR     R0              ;DURATION OF EACH CONVERT
010522  005200                  4$:     INC     R0              ;LOOP TIME FOR EACH CONVERT HERE
010524  001376                          BNE     4$
010526  005201                          INC     R1              ;COUNT # OF CONVERTS HERE
010530  001373                          BNE     3$
010532  052715  020000                  BIS     #20000,@ADCSR5  ;REINIT THE A/D
010536  104400                          ERROR                   ;DONE BIT NOT SET
010540  000401                          BR      SYNRE3+2
                        ;TEST THAT THE DATA BUFFER HAS BEEN MODIFIED

010542  022626                  SYNRE3: POP2SP                  ;POP 2 FROM STACK
010544  005715                          TST     @ADCSR5         ;TEST FOR A STATUS ERROR
010546  100001                          BPL     .+4
010550  104400                          ERROR                   ;INTERRUPT WAS FROM ADF STATUS ERROR
010552  012701  015072                  MOV     #ADBUFF,R1      ;GET DATA BUFFER POINTER
010556  022721  152525          6$:     CMP     #152525,(R1)+   ;VERIFY DATA WAS MODIFIED
010562  001404                          BEQ     7$              ;IF SAME DATA WASN'T MODIFIED
010564  022701  015272                  CMP     #ADBUFF+200,R1  ;DONE YET
010570  001372                          BNE     6$
010572  000401                          BR      8$
010574  104400                  7$:     ERROR                   ;DATA BUFFER WASNT MODIFIED (DONE BIT SET)
010576  104002                  8$:     SCOPE
010600  104000                          PRINT                   ;PRINT EXT. INT. TEST COMPLETE
010602  013710                          MES42
010604  000167  177224                  JMP     CYC             ;RECYCLE TEST
```

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

010610  004767  171112        TESTX:  JSR     PC,SETUP        ;MAIN INITIALIZATION FOR LOGIC TEST
010614  005067  000444                CLR     INBUF
010620  012701  011264                MOV     #INBUF,R1
010624  005067  003210                CLR     KSTOR1
010630  042711  177770        TSTA:   BIC     #177770,(R1)    ;STRIPE NO. TO OCTAL
010634  062167  003200                ADD     (R1)+,KSTOR1    ;ADD TO LAST RESULT
010640  005367  003170                DEC     CHRCNT
010644  001407                        BEQ     TSTB
010646  006367  003166                ASL     KSTOR1
010652  006367  003162                ASL     KSTOR1
010656  006367  003156                ASL     KSTOR1
010662  000762                        BR      TSTA

010664  022767  010610  003146  TSTB:   CMP     #TESTX,KSTOR1   ;IS NO. WITHIN LIMITS OF THE LOGIC TEST
010672  100002                        BPL     .+6             ;CONTINUE IF YES
010674  000167  171020                JMP     INIT3           ;OTHERWISE RETURN TO MONITOR
010700  062767  000002  003132        ADD     #2,KSTOR1       ;ADD '2' TO POINT TO INSTRUCTION AFTER SCOPE
010706  005067  001774        XLOOP:  CLR     SCOPEF          ;KEEP COUNT AT ZERO
010712  012767  010706  001770        MOV     #XLOOP,RETURN   ;LOAD SCOPE LOOP RETURN POINTER
010720  000177  003114                JMP     @KSTOR1         ;JUMP TO TEST

                                ;SUBROUTINE TO ISSUE N SPACES
                                ;N IS ONE PLUS VALUE CONTAINED IN SPACEX
                                ;SPACEX IS CLEARED WITHIN THE SUBROUTINE, SO THAT A CALL ON
                                ;SPACE WITHOUT LOADING SPACEX ISSUES ONLY ONE SPACE

010724  105777  170346        XSPACE: TSTB    @TPS            ;WAIT FOR TTY READY
010730  100375                        BPL     .-4
010732  012777  000240  170340        MOV     #240,@TPB       ;OUTPUT A SPACE
010740  005367  000010                DEC     SPACEX          ;DECREMENT COUNT
010744  003367                        BGT     XSPACE          ;LOOP IF NOT DONE
010746  005067  000002                CLR     SPACEX          ;RESET COUNT TO ZERO
010752  000002                        RTI                     ;RETURN
010754  000000        SPACEX: 0

                                ;SUBROUTINE TO SET UP ALL A/D REGISTERS

010756  012777  014070  170330  SETREG: MOV     #RANBUF,@ADSWR
010764  012777  015072  170326          MOV     #ADBUFF,@ADWRA
010772  017777  170314  170322          MOV     @ADWCR,@ADWRB
011000  005477  170316                  NEG     @ADWRB
011004  006377  170312                  ASL     @ADWRB
011010  062777  015072  170304          ADD     #ADBUFF,@ADWRB
011016  000207                        RTS     PC
```

```
                                        ;KEYBOARD SERVICE ROUTINE

011020  104012                  XTTYIN: SAVREG
011022  012704  011264                  MOV     #INBUF,R4       ;SETUP CHARACTER BUFFER
011026  005067  003002                  CLR     CHRCNT          ;CLEAR CHARACTER COUNTER
011032  005067  000230                  CLR     INBUF+2
011036  105777  170230          INPUTA: TSTB    @TKS            ;CHARACTER READY?
011042  100375                          BPL     INPUTA          ;NO, WAIT IT OUT
011044  017701  170224                  MOV     @TKB,R1         ;SAVE CHARACTER
011050  042701  000200                  BIC     #200,R1         ;STRIPE PARITY BIT
011054  120127  000060                  CMPB    R1,#60          ;IS IT A SPECIAL CHARACTER
011060  100420                          BMI     SPCHR1          ;YES, TEST IT
011062  122701  000137                  CMPB    #137,R1
011066  100415                          BMI     SPCHR1
011070  010124                  INPUTB: MOV     R1,(R4)+        ;SAVE CHARACTER
011072  005267  002736                  INC     CHRCNT          ;INCREMENT THE CHARACTER COUNT.
011076  022767  000007  002730          CMP     #7,CHRCNT
011104  100461                          BMI     SPCHR5          ;TYPE '?' IF TOO MANT CHAR.
011106  105777  170164          OUTPTA: TSTB    @TPS            ;ECHO CHARACTER
011112  100375                          BPL     OUTPTA
011114  110177  170160                  MOVB    R1,@TPB
011120  000746                          BR      INPUTA          ;WAIT FOR NEXT CHARACTER
                                        ;SUBROUTINE TO TEST FOR SPECIAL CHARACTERS :'↑C','+','CR',',' OR 'RUBOUT'

011122  122701  000003          SPCHR1: CMPB    #3,R1           ;CHAR.='↑C'
011126  001002                          BNE     .+6             ;NO, NOT '↑C'
011130  000167  170446                  JMP     MONITR          ;YES, EXIT TO MONITOR
011134  122701  000001                  CMPB    #1,R1           ;CHAR. ='↑A'?
011140  001006                          BNE     .+16            ;NOT '↑A'
011142  022626                          POP2SP                  ;YES, RESTORE STACK
011144  104000                          PRINT
011146  013306                          CNTRLA                  ;TEXT '↑A'
011150  104013                          GETREG                  ;RESTORE THE REG.'S
011152  000177  002626                  JMP     @AVECTR
011156  122701  000177                  CMPB    #177,R1         ;CHAR. = 'RUBOUT'
011162  001011                          BNE     SPCHR2          ;IGNORE CHAR. & EXIT
011164  005767  002644                  TST     CHRCNT          ;IS RUBOUT LEGAL?
011170  001722                          BEQ     INPUTA          ;NO, IGNORE IT
011172  005367  002636                  DEC     CHRCNT
011176  012701  000134                  MOV     #134,R1         ;TYPE '\' TO INDICATE RUBOUUT
011202  005744                          TST     -(R4)           ;POP OFF LAST CHARACTER
011204  000740                          BR      OUTPTA          ;WAIT FOR NEXT CHARACTER
011206  122701  000053          SPCHR2: CMPB    #53,R1          ;TEST FOR '+'
011212  001004                          BNE     SPCHR3          ;BRANCH IF NO
011214  012767  000177  002550          MOV     #177,ADSIGN     ;YES, IDICATES UNIPOLAR
011222  000731                          BR      OUTPTA          ;WAIT NEXT CHAR.
011224  122701  000054          SPCHR3: CMPB    #54,R1          ;TEST FOR ','
011230  001717                          BEQ     INPUTB          ;LEGAL CHAR., SAVE IT
011232  122701  000015          SPCHR4: CMPB    #15,R1          ;=TO 'CARRIAGE RETURN' TO TERMINATE?
011236  001004                          BNE     SPCHR5          ;NO, CONTINUE
011240  104000                          PRINT                   ;YES, TYPE 'CR-LF'
011242  013311                          CRLF
011244  104013                  EXTTY:  GETREG                  ;RESTORE THE REG.'S
011246  000002                          RTI                     ;EXIT
```

```
011250  122701  000040          SPCHRS: CMPB    #40,R1          ;TEST FOR SPACE
011254  001714                          BEQ     OUTPTA          ;ECHO BUT DON'T SAVE
011256  104000                          PRINT                   ;OTHERWISE TYPE '?'
011260  013316                          QMARK
011262  000657                          BR XTTYIN+2             ;WAIT FOR NEW ENTRY
011264  000000                  INBUF:  0                       ;CHARACTER STORAGE BUFFER
        011304                          .=.+16
                                ;SUBROUTINE WILL CONVERT 'N' BCD WORDS (SEPARATED VIA COMMA'S)
                                ;WHICH WERE STORED IN A TABLE VIA 'TTYIN' TO OCTAL AND STORE THEM.

011304  104012                  BCDBIN: SAVREG                  ;SAVE THE REG.'S
011306  012704  011264                  MOV     #INBUF,R4       ;SETUP ASCII STORAGE TABLE
011312  012703  011460                  MOV     #BCDTAB,R3      ;TABLE FOR STORAGE OF CONVERTED WORDS
011316  005067  000140                  CLR     BCDTAB+2
011322  005001                  BCDBN1: CLR     R1              ;REG. TO STORE RUNNING TOTAL
011324  005002                          CLR     R2              ;TEMP. STORAGE FOR 'R1'
011326  005767  002502          BCDBN2: TST     CHRCNT          ;END OF DATA?
011332  003426                          BLE     BCDEND          ;YES, EXIT
011334  005367  002474                  DEC     CHRCNT          ;DECREMENT CHARACTER COUNTER
011340  122714  000054                  CMPB    #54,(R4)        ;IS CHARACTER = TO ' '?
011344  001421                          BEQ     BCDEND          ;YES, DECODE NEW WORD
011346  121427  000060                  CMPB    (R4),#60
011352  002435                          BLT     BCDERR          ;TEST FOR LEGAL NO.
011354  021427  000071                  CMP     (R4),#71
011360  003032                          BGT     BCDERR
011362  042714  177760                  BIC     #177760,(R4)    ;STRIPE NO. TO BCD
011366  012400                          MOV     (R4)+,R0        ;SAVE NO. IN R0.
011370  010102                          MOV     R1,R2           ;SAVE CURRENT TOTAL
011372  006301                          ASL     R1              ;X2
011374  006301                          ASL     R1              ;X4
011376  006301                          ASL     R1              ;X8
011400  060201                          ADD     R2,R1           ;X9
011402  060201                          ADD     R2,R1           ;X10
011404  060001                          ADD     R0,R1           ;N+NEW NO.
011406  000747                          BR      BCDBN2
011410  005724                  BCDEND: TST     (R4)+           ;UPDATE BUFFER
011412  010123                          MOV     R1,(R3)+        ;SAVE CONVERTED VALUE & SETUP TO SAVE NEXT
011414  005767  002414                  TST     CHRCNT          ;FINISHED?
011420  001340                          BNE     BCDBN1          ;NO, CONVERT NEXT WORD
011422  026727  000032  000777          CMP     BCDTAB,#777     ;TEST IF NO. <511
011430  100006                          BPL     BCDERR          ;REPORT ERROR IF NOT
011432  026727  000024  000777          CMP     BCDTAB+2,#777   ;TEST IF 2ND. NO. <511
011440  100002                          BPL     BCDERR          ;BRANCH IF NOT
011442  104013                          GETREG                  ;RESTORE THE REG.'S
011444  000002                          RTI                     ;YES, EXIT
011446  104000                  BCDERR: PRINT
011450  013316                          QMARK                   ;TYPE '?'.
011452  104005                          TTYIN                   ;TO BE TYPED ON QUESTIONABLE ENTRIES.
011454  000167  177624                  JMP     BCDBIN
011460  000000                  BCDTAB: 0                       ;OCTAL STORAGE TABLE
011462  000000                          0
011464  000000                          0
011466  000000                          0
```

```
                                                ;POWER FAIL HANDLER
011470  010046                          PWRFAL: MOV     R0,-(SP)
011472  010146                                  MOV     R1,-(SP)
011474  010246                                  MOV     R2,-(SP)
011476  010346                                  MOV     R3,-(SP)
011500  010446                                  MOV     R4,-(SP)
011502  010546                                  MOV     R5,-(SP)
011504  016746  166314                         MOV     24,-(SP)
011510  010667  002316                         MOV     SP,PROC
011514  012767  011524  166302                 MOV     #PWRUP,24
011522  000000                                  HALT
                                                ;POWER UP HANDLER
011524  012777  000340  167536          PWRUP:  MOV     #340,@PSW
011532  016706  002274                         MOV     PROC,SP
011536  012667  166262                         MOV     (SP)+,24
011542  012605                                  MOV     (SP)+,R5
011544  012604                                  MOV     (SP)+,R4
011546  012603                                  MOV     (SP)+,R3
011550  012602                                  MOV     (SP)+,R2
011552  012601                                  MOV     (SP)+,R1
011554  012600                                  MOV     (SP)+,R0
011556  005001                                  CLR     R1              ;POWER UP DELAY
011560  005201                                  INC     R1
011562  001376                                  BNE     .-2
011564  104000                                  PRINT
011566  013617                                  MES21
011570  000167  170006                         JMP     MONITR

                                                ;*****************************************************************************
                                                ;THIS SUBROUTINE IS CALLED VIA A 'IOT' TRAP.  IT IS USED BY MANY LOGIC
                                                ;SUBTESTS TO CHECK FOR 'A/D DONE'.  THE ROUTINE CHECKS FOR 'A/D DONE'
                                                ;WHILE INCREMENTING A WAIT LOOP.  IF 'DONE' SETS PROGRAM CONTROL IS THEN
                                                ;TRANSFERRED BACK TO THE 'IOT' CALL +4.  IF THE LOOP TIMES OUT (NO DONE
                                                ;SETS), PROGRAM CONTROL IS TRANSFERRED BACK TO THE 'IOT' CALL+2
                                                ;*****************************************************************************

011574  016701  002176          XCKDON: MOV     DELAY1,R1       ;INITIALIZE WAIT LOOP
011600  105715                  XCHK1:  TSTB    @ADCSR5         ;TEST FOR DONE
011602  100403                          BMI     EXTCHK          ;BRANCH IF SET
011604  005201                          INC     R1              ;INCREMENT WAIT LOOP
011606  001374                          BNE     XCHK1
011610  000002                          RTI                     ;RETURN TO CALL+2
011612  062716  000002          EXTCHK: ADD     #2,(SP)         ;RETURN TO CALL +4
011616  000002                          RTI
```

```
                                ;SUBROUTINE TO SET UP THE A/D VECTOR ADDR TO ENABLE INTERRUPTS.

011620  017677  000000  167500  LDINTR: MOV     @(SP),@ADINT    ;LOAD INTERRUPT SERVICE ADDRESS
011626  062716  000002                  ADD     #2,(SP)         ;SET UP STACK TO EXIT
011632  012777  000340  167470          MOV     #340,@ADLVL     ;SET A/D LEVEL @ 7
011640  052777  000100  167442          BIS     #100,@ADCSR     ;SET INTERRUPT ENABLE
011646  005077  167416                  CLR     @PSW            ;SET PROC. @ 0
011652  000207                          RTS     PC

                                ;SUBROUTINE TO RESET A/D VECTOR ADDRESS TO HALT ON INTERRUPTS

011654  012777  000340  167406  CLRINT: MOV     #340,@PSW       ;RE-SET PROC. PRIORITY @ 7
011662  042777  000100  167420          BIC     #100,@ADCSR     ;CLR INTR ENABLE
011670  016777  167434  167430          MOV     ADLVL,@ADINT
011676  005077  167426                  CLR     @ADLVL
011702  000207                          RTS     PC

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

011704  011667  002140          ERTRAP: MOV     (SP),TEMP1      ;SAVE LOCATION WHERE IT TRAPPED 'TO'
011710  022626                          POP2SP
011712  011667  002134                  MOV     (SP),TEMP2      ;SAVE WHERE IT TRAPPED FROM.
011716  104000                          PRINT
011720  013657                          MES40                   ;TEXT 'ILLEGAL TRAP TO'
011722  162767  000004  002120          SUB     #4,TEMP1
011730  104004                          PRTOCT
011732  014050                          TEMP1                   ;TYPE 'PC' TRAPPED TO
011734  104000                          PRINT
011736  013701                          MES41                   ;TEXT 'FROM'
011740  162767  000002  002104          SUB     #2,TEMP2
011746  104004                          PRTOCT
011750  014052                          TEMP2                   ;TYPE WHERE IT TRAPPED FROM
011752  000167  167624                  JMP     MONITR          ;RETURN TO MONITOR
                                ;*****************************************************************
                                ;ROUTINE TO INITIALIZE THE A/D
                                ;*****************************************************************

011756  052777  020000  167324  XINIT:  BIS     #20000,@ADCSR
011764  000002                          RTI
```

```
                                ;*****************************************************************************
                                ;SUBROUTINE TO SAVE 'R1-R5' ON STACK
                                ;*****************************************************************************

011766  012667  002016         XSAVRG: MOV     (SP)+,SAVEPC
011772  012667  002014                 MOV     (SP)+,SAVPSW
011776  012667  002012                 MOV     (SP)+,SAV2PC
012002  012667  002010                 MOV     (SP)+,SAV2SW
012006  010146                         MOV     R1,-(SP)
012010  010246                         MOV     R2,-(SP)
012012  010346                         MOV     R3,-(SP)
012014  010446                         MOV     R4,-(SP)
012016  010546                         MOV     R5,-(SP)
012020  016746  001772                 MOV     SAV2SW,-(SP)
012024  016746  001764                 MOV     SAV2PC,-(SP)
012030  016746  001756                 MOV     SAVPSW,-(SP)
012034  016746  001750                 MOV     SAVEPC,-(SP)
012040  000002                         RTI

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

012042  012667  001742         XGETRG: MOV     (SP)+,SAVEPC
012046  012667  001740                 MOV     (SP)+,SAVPSW
012052  012667  001736                 MOV     (SP)+,SAV2PC
012056  012667  001734                 MOV     (SP)+,SAV2SW
012062  012605                         MOV     (SP)+,R5
012064  012604                         MOV     (SP)+,R4
012066  012603                         MOV     (SP)+,R3
012070  012602                         MOV     (SP)+,R2
012072  012601                         MOV     (SP)+,R1
012074  016746  001716                 MOV     SAV2SW,-(SP)
012100  016746  001710                 MOV     SAV2PC,-(SP)
012104  016746  001702                 MOV     SAVPSW,-(SP)
012110  016746  001674                 MOV     SAVEPC,-(SP)
012114  000002                         RTI
```

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

012116  104012                  TYPMES: SAVREG                  ;SAVE THE REGISTERS
012120  017602  000000                  MOV     @(SP),R2        ;GET THE MESSAGE ADDRESS FROM START
012124  062716  000002                  ADD     #2,(SP)         ;SET UP STACK TO EXIT
012130  012777  000100  167134          MOV     #100,@TKS
012136  005077  167126                  CLR     @PSW            ;ENABLE KEYBOARD INTR.
012142  105777  167130          TYPERA: TSTB    @TPS
012146  100375                         BPL     TYPERA          ;WAIT FOR TTY DONE
012150  122712  000100                  CMPB    #100,(R2)       ;TEST FOR '@'
012154  001002                         BNE     TYPER1          ;BRANCH IF NO EQUAL
012156  104013                         GETREG
012160  000002                         RTI                     ;OTHERWISE EXIT
012162  122712  000045          TYPER1: CMPB    #45,(R2)        ;TEST FOR '%'
012166  001403                         BEQ     TYPECL          ;IF = TYPE 'CR-LF'
012170  112277  167104          TYPER2: MOVB    (R2)+,@TPB      ;OUTPUT CHAR.
012174  000762                         BR      TYPERA
012176  012777  000015  167074  TYPECL: MOV     #15,@TPB        ;TYPE 'CR'
012204  105777  167066                  TSTB    @TPS
012210  100375                         BPL     .-4
012212  012777  000012  167060          MOV     #12,@TPB
012220  105722                         TSTB    (R2)+           ;INCREMENT BUFFER
012222  000747                         BR      TYPERA

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

012224  104012                  OCTPRT: SAVREG
012226  012777  000100  167036          MOV     #100,@TKS
012234  005077  167030                  CLR     @PSW            ;ENABLE KEYBOARD INTR.
012240  017600  000000                  MOV     @(SP),R0        ;THE ADDRESS OF WORD TO BE TYPED
012244  062716  000002                  ADD     #2,(SP)         ;SET UP STACK TO EXIT
012250  012767  000006  001564          MOV     #6,KSTOR2
012256  012767  000376  000056          MOV     #376,MASK       ;MASK FOR FIRST BIT
012264  000401                         BR      .+4
012266  006110                  SHIFT:  ROL     (R0)
012270  006110                          ROL     (R0)
012272  006110                          ROL     (R0)
012274  111002                          MOVB    (R0),R2
012276  146702  000040                  BICB    MASK,R2
012302  052702  000260                  BIS     #260,R2
012306  132777  000200  166762          BITB    #200,@TPS
012314  100374                         BPL     .-6             ;WAIT FOR PRINTER READY
012316  110277  166756                  MOVB    R2,@TPB         ;PRINT CHAR.
012322  012767  000370  000012          MOV     #370,MASK       ;MASK FOR NEXT '5' DIGITS
012330  005367  001506                  DEC     KSTOR2
012334  001354                         BNE     SHIFT
012336  104013                         GETREG
012340  000002                         RTI
012342  000376                  MASK:   376
```

```
                                ;ENTERED WITH SYSTEM TRAP CALL (ERROR)
                                ;PRINT OUT THE ERROR ADDRESS AND ALL A/D STATUS REGISTERS

012344  104006                LOGERR: TSTTKS                  ;TEST FOR KEYBOARD INTERRUPT
012346  037727  166730  020000        BIT     @SWR,#20000     ;TEST SW-13 FOR INHIBIT PRINT OUT
012354  001110                        BNE     CK              ;INHIBIT,CHECK FOR HALT
012356  011667  001462                MOV     (SP),KSTOR3     ;PC OF FAILING ROUTINE
012362  162767  000002  001454        SUB     #2,KSTOR3
012370  042777  000100  166712        BIC     #100,@ADCSR     ;CLR INTERRUPT ENABLE
012376  010046                        MOV     R0,-(SP)        ;SAVE REGISTERS 0,1 &2
012400  010146                        MOV     R1,-(SP)
012402  010246                        MOV     R2,-(SP)
012404  022767  007662  001372        CMP     #DATA1,AVECTR           ;RUNNING DATA TEST ?
012412  001034                        BNE     ERR1            ;NO,LOGIC ERROR
012414  005701                        TST     R1
012416  001003                        BNE     .+10
012420  012767  013547  000046        MOV     #MES10,PRNTIT
012426  022701  000001                CMP     #1,R1
012432  001003                        BNE     .+10
012434  012767  013561  000032        MOV     #MES11,PRNTIT
012442  022701  000002                CMP     #2,R1
012446  001003                        BNE     .+10
012450  012767  013573  000016        MOV     #MES12,PRNTIT
012456  022701  000003                CMP     #3,R1
012462  001003                        BNE     .+10
012464  012767  013605  000002        MOV     #MES13,PRNTIT
012472  104000                        PRINT
012474  000000                PRNTIT: 0
012476  104004                        PRTOCT
012500  014046                        KSTOR4                  ;PRINT READ DATA
012502  000432                        BR      XPRT1           ;EXIT
012504  005767  001346        ERR1:   TST     MESPRT          ;HAS HEADER BEEN TYPED
012510  001004                        BNE     ERR2            ;BRANCH IF YES
012512  104000                        PRINT
012514  013101                        MES3                    ;PRINT LOGIC ERROR HEADER
012516  005267  001334                INC     MESPRT          ;SET PRINT INHIBIT SW.
012522  104000                ERR2:   PRINT                   ;OUTPUT CARRIAGE RETURN AND LINE FEED
012524  013311                        CRLF
012526  104004                        PRTOCT                  ;PRINT FAILING PC+2
012530  014044                        KSTOR3
012532  104003                        SPACE                   ;OUTPUT A SPACE
012534  012767  000007  001306        MOV     #7,TEMP1
012542  012701  001306                MOV     #ADCR,R1
012546  012100                XPRT:   MOV     (R1)+,R0
012550  011067  001270                MOV     (R0),KSTOR3
012554  104004                        PRTOCT
012556  014044                PRTADR: KSTOR3
012560  104003                        SPACE
012562  005367  001262                DEC     TEMP1
012566  001367                        BNE     XPRT
```

```
012570  012602                  XPRT1:  MOV     (SP)+,R2        ;RESTORE REGISTERS
012572  012601                          MOV     (SP)+,R1
012574  012600                          MOV     (SP)+,R0
012576  005777  166500          CK:     TST     @SWR            ;CHECK SW-15 FOR HALT SWITCH
012602  100001                          BPL     .+4             ;BRANCH IF NOT SET
012604  000000                          HALT                    ;HALT ON ERROR UP
012606  032777  010000  166466          BIT     #SW12,@SWR      ;TEST SW 12
012614  001401                          BEQ     .+4             ;BRANCH IF NOT SET
012616  104011                          INITAD                  ;CALL ROUTINE TO INITIALIZE THE 'A/D'
012620  000002                          RTI                     ;RETURN TO MAIN LINE

                                ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 200 TIMES

012622  104006                  SCOPEC: TSTTKS                  ;TEST FOR KEYBOARD IMIT
012624  032777  040000  166450          BIT     #40000,@SWR     ;TEST SW-14 FOR SCOPE
012632  001012                          BNE     SCOPEB          ;YES, SCOPE
012634  032777  004000  166440          BIT     #4000,@SWR      ;NO-TEST SW-11 FOR ITERATION
012642  001013                          BNE     SCOPEG          ;INHIBIT ITERATION
012644  026767  000036  000032          CMP     SCOPEF,ICOUNT   ;COMPARE CURRENT COUNT TO MAX NUMBER
012652  100007                          BPL     SCOPEG          ;EXIT-DONE
012654  005267  000026                  INC     SCOPEF          ;INCREMENT COUNT
012660  022606                  SCOPEB: CMP     (6)+,SP         ;REPOSITION STACK
012662  012677  166402                  MOV     (6)+,@PSW       ;RESTORE PREVIOUS PROCESSOR STATUS
012666  000177  000016                  JMP     @RETURN         ;REPEAT TEST
012672  005067  000010          SCOPEG: CLR     SCOPEF          ;CLEAR COUNT
012676  011667  000006                  MOV     @SP,RETURN      ;SAVE SCOPE RETURN POINTER
012702  000002                          RTI                     ;RETURN INLINE-NEXT TEST
012704  000200                  ICOUNT: 200                     ;ITERATION COUNT
012706  000000                  SCOPEF: 0                       ;COUNT LOCATION FOR ITERATION LOOP
012710  002014                  RETURN: TST1

                                ;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET

012712  105777  166354          TKSFLG: TSTB    @TKS            ;FLAG SET?
012716  100001                          BPL     .+4             ;NO, EXIT
012720  104005                          TTYIN                   ;YES,INQUIRE
012722  000002                          RTI

                                ;*****************************************************************
                                ;ROUTINE TO TRANSMIT A 'NULL' CHAR. TO THE PRINTER
                                ;*****************************************************************

012724  012767  000002  176022  XNULL:  MOV     #2,SPACEX
012732  105777  166340                  TSTB    @TPS
012736  100375                          BPL     .-4
012740  005077  166334                  CLR     @TPB            ;TRANSMIT NULL CHAR.
012744  005367  176004                  DEC     SPACEX
012750  001370                          BNE     XNULL+6
012752  000002                          RTI
```

```
                                        ;MESSAGES
012754      000                                 .BYTE
012755      045  040445  043104  TITLE:  .ASCII  '%%ADF11 PART I, LOGIC DIAGNOSTIC TEST, 17-JUL-75'
012762  030461  050040  051101
012770  020124  026111  046040
012776  043517  041511  042040
013004  040511  047107  051517
013012  044524  020103  042524
013020  052123  020054  033461
013026  045055  046125  033455
013034      065
013035      040  046450  044501          .ASCII  ' (MAINDEC-11-DZADG-A)@'
013042  042116  041505  030455
013050  026461  055104  042101
013056  026507  024501     100

013063      045  027501  020104  MES2:   .ASCII  '%A/D LENGTH? @'
013070  042514  043516  044124
013076  020077     100
013101      045  020040  041520  MES3:   .ASCII  '%  PC     CR    CSR     WC   STATUS  DATA  '
013106  020040  020040  041440
013114  020122  020040  041440
013122  051123  020040  020040
013130  053440  020103  020040
013136  052123  052101  051525
013144  020040  040504  040524
013152  020040
013154  042101  051104  040440          .ASCII  'ADDR A ADDR B@'
013162  040440  042104  020122
013170  040102

013172  052045  050131  020105  MES4:   .ASCII  "%TYPE LETTER ( ) TO RUN DESIRED TEST: "
013200  042514  052124  051105
013206  024040  024440  052040
013214  020117  052522  020116
013222  042504  044523  042522
013230  020104  042524  052123
013236  020072
013240  046050  047451  044507          .ASCII  "(L)OGIC, (D)DATA, (S)SYNC (T)NNN%@"
013246  026103  024040  024504
013254  040504  040524  020054
013262  051450  051451  047131
013270  020103  052050  047051
013276  047116  040045

013302  041536  040045          CNTRLC: .ASCII  '↑C%@'
013306  040536     100          CNTRLA: .ASCII  '↑A@'

013311      045     100          CRLF:   .ASCII  '%@'

013313      045  040056          DOT:    .ASCII  '%.@'

013316  020077     100          QMARK:  .ASCII  '? @'

013321      045  046042  043517  MES5:   .ASCII  '%"LOGIC TEST"%@'
```

```
013326  041511  052040  051505
013334  021124  040045
013340  021045  042524  052123  MES6:   .ASCII  ;%"TEST COMPLETE"@;
013346  041440  046517  046120
013354  052105  021105     100
013361     045  042042  052101  MES7:   .ASCII  '%"DATA UPDATE TEST"@'
013366  020101  050125  040504
013374  042524  052040  051505
013402  021124     100
013405     123  050125  046120          .ASCII  ;SUPPLY THE FOLLOWING VOLTAGES TO CH. '0'.;
013412  020131  044124  020105
013420  047506  046114  053517
013426  047111  020107  047526
013434  052114  043501  051505
013442  052040  020117  044103
013450  020056  030047  027047
013456  054524  042520  023440  MES44:  .ASCII  ;TYPE 'CR' TO START TEST.%@;
013464  051103  020047  047524
013472  051440  040524  052122
013500  052040  051505  027124
013506  040045
013510  051445  050125  046120  MES8:   .ASCII  "%SUPPLY '+5V' !@"
013516  020131  025447  053065
013524  020047  040041
013530  052523  050120  054514  MES9:   .ASCII  "SUPPLY '-5V' !@"
013536  023440  032455  023526
013544  020440     100
013547     045  051461  020124  MES10:  .ASCII  '%1ST WRD @'
013554  051127  020104     100
013561     045  047062  020104  MES11:  .ASCII  '%2ND WRD @'
013566  051127  020104     100
013573     045  051063  020104  MES12:  .ASCII  '%3RD WRD @'
013600  051127  020104     100
013605     045  052064  020110  MES13:  .ASCII  '%4TH WRD @'
013612  051127  020104     100

013617     045  051045  041505  MES21:  .ASCII  '%%RECOVERED FROM POWER FAILURE @'
013624  053117  051105  042105
013632  043040  047522  020115
013640  047520  042527  020122
013646  040506  046111  051125
013654  020105     100
013657     045  046111  042514  MES40:  .ASCII  ;%ILLEGAL TRAP TO @;
013664  040507  020114  051124
013672  050101  052040  020117
013700     100

013701     040  051106  046517  MES41:  .ASCII  ; FROM @;
013706  040040
013710  054105  027124  044455  MES42:  .ASCII  ;EXT.-INT. TEST COMPLETE %@;
013716  052116  020056  042524
013724  052123  041440  046517
013732  046120  052105  020105
013740  040045
013742  047503  053116  051105  MES43:  .ASCII  ;CONVERTION SYNC TEST %@;
013750  044524  047117  051440
```

```
013756  047131  020103  042524
013764  052123  022440     100

013772                          .EVEN
```

```
                                ;ADDRESS AND CONSTANTS TABLE

013772  000000  ADSIGN: 0                ;UNIPOLAR=0, BIPOLAR=1
013774  000000  SIGNBF: 0                ;A/D WORD LENGTH
013776  177700  DELAY1: -100             ;DELAY COUNT FOR LOCIC TEST
014000  177634  DELAY2: -100.            ;DELAY COUNT FOR LOGIC TEST
014002  001000  STACK:  1000             ;INITIAL SP. ADDRESS
014004  001546  AVECTR: INITB            ;'↑A' VECTOR ADDRESS
014006  001720  PVECTR: INIT3            ;'↑P' VECTOR ADDRESS
014010  000000  SAVEPC: 0
014012  000000  SAVPSW: 0
014014  000000  SAV2PC: 0
014016  000000  SAV2SW: 0
014020  000000  INCFLG: 0                ;SOFTWARE FLAG;0=NO INC. MEM.
014022  000000  MEMSIZ: 0                ;CALCULATED MEM SIZE TO SUPPORT INC. MEM.
014024  000000  CORMAX: 0                ;CALCULATED MEMORY SIZE
014026  000000  SOFLAG: 0                ;SOFTWARE 'FLAG'
014030  000000  ADSIZE: 0                ;OCTAL STORAGE OF A/D LENGTH
014032  000000  PROC:   0                ;TEMP STORAGE FOR 'PSW'
014034  000000  CHRCNT: 0                ;TEMP STORAGE
014036  000000  COUNT:  0                ;TEMP STORAGE
014040  000000  KSTOR1: 0                ;PERMANENT STORAGE
014042  000000  KSTOR2: 0                ;PERMANENT STORAGE
014044  000000  KSTOR3: 0                ;PERMANENT STORAGE
014046  000000  KSTOR4: 0                ;PERMANENT STORAGE
014050  000000  TEMP1:  0                ;TEMPORARY STORAGE
014052  000000  TEMP2:  0                ;TEMPORARY STORAGE
014054  000000  TEMP3:  0                ;TEMPORARY STORAGE
014056  000000  MESPRT: 0
014060  000000  HIGH:   0
014062  000000  LOW:    0
014064  000000  GM07XB: 0
014066  000000  GM03XB: 0

                                ;HERE STARTS THE '1000' WORD STATUS WORD BUFFER
014070  000000  RANBUF: 0
        015072          .=.+1000
                                ;HERE STARTS THE WORD A/D DATA BUFFER.
015072  000000  ADBUFF: 0

        001332          .END    INIT
```

```
ADADR   001324   685# 1974* 1998*
ADBUFF  015072  1540  1554  1581  1594  1605  1619  1632* 1645  1652  1662  1781  1785  1806
                1815  1864  1895  1920  1930  1948  2110  2112  2129  2134  2277  2279  2299
                2315  2317  2340  2343  2354  2356  2384  2387  2437  2441  2982#
ADCR    001306   678# 787   2769
ADCR4 =X000004   338# 787*  879*  889*  920*  921   930*  931*  932   941*  942*  943   951*
                 964* 994*  1155* 1156* 1186* 1189* 1199* 1202* 1216* 1230* 1240* 1251* 1263*
                1295* 1324* 1333* 1334* 1335* 1336* 1340  1347  1354  1361  1372* 1378* 1379*
                1380* 1381* 1382* 1391  1399* 1400* 1415* 1416* 1424  1431* 1432* 1451* 1452*
                1462  1472* 1473* 1483  1493* 1497  1510* 1511* 1522* 1523* 1536* 1537* 1550*
                1551* 1565* 1566* 1586* 1587* 1626* 1628* 1678* 1680* 1690  1708* 1709* 1735*
                1752* 1755  1774* 1789  1812* 1834* 1887* 1977* 1978* 2005* 2031* 2033* 2077*
                2118* 2120* 2190* 2259* 2260* 2286* 2287* 2327* 2368*
ADCSR   001310   679# 788   2600* 2607* 2635* 2735*
ADCSR5=X000005   339# 788*  807   871*  872   880*  881   890*  892   900*  902   911*  912
                 929* 952*  953   954   965*  966*  967   977*  978   986*  987   995*  996
                1019* 1033* 1046* 1064* 1068* 1083* 1091  1102* 1118* 1133* 1146* 1147  1165*
                1173* 1174  1187* 1188* 1200* 1201* 1205* 1214* 1215* 1226* 1227* 1231* 1254
                1257* 1268  1274  1280  1286* 1287  1301  1305  1314  1327* 1375* 1407  1420*
                1436  1439* 1442  1459* 1461  1480* 1521* 1535* 1549* 1564* 1585* 1627* 1637*
                1638  1679* 1685* 1688  1704* 1716  1753  1775  1813  1835  1888* 1894* 1907*
                1919* 1937* 1944* 2012  2032* 2036* 2042* 2080* 2119* 2126* 2191  2193* 2264*
                2272  2296  2337  2375* 2381  2588
ADDBR   001316   682# 786   1176
ADDBR3=X000003   337# 786*  891   901   935   946   958   970   999   1159  1168  1180  1193
                1209  1235  1244  1304  1317  1343  1350  1357  1385  1403  1419  1423  1435
                1445  1456  1477  1496  2175  2179
ADINT   001326   686# 1905* 1935* 2597* 2608*
ADLVL   001330   687# 1906* 1936* 2599* 2606  2609*
ADSIGN  013772   699* 722   2487* 2946#
ADSIZE  014030   732* 2961#
ADSWR   001314   681# 814   862*  863   1005* 1006* 1007  1569  1777  1837  1981  2008  2060*
                2065  2436*
ADWCR   001312   680# 853*  854   1013* 1014* 1015  1018* 1020  1057* 1228* 1508* 1519* 1526
                1533* 1547* 1562* 1578* 1616* 1674* 1702* 1733* 1750* 1772* 1810* 1832* 1868*
                1975* 2003* 2020* 2059* 2114* 2255* 2283* 2325* 2366* 2438
ADWRA   001320   683# 821   835*  836   1027* 1028* 1029  1032* 1034  1540  1705* 1781  2021*
                2061* 2062* 2064  2070  2083  2437*
ADWRB   001322   684# 828   844*  845   1040* 1041* 1042  1045* 1047  1554  1618* 1785  1870*
                2022* 2023* 2063* 2064* 2438* 2439* 2440* 2441*
AVECTR  014004   761* 2158* 2476  2739  2951#
BCDBIN  011304   658  2507#
BCDBN1  011322  2511#
BCDBN2  011326  2513#
BCDEND  011410  2514  2517  2532#
BCDERR  011446  2519  2521  2537  2539  2542#
BCDTAB  011460   706* 711*  2509  2510* 2536  2538  2546#
CHRCNT  014034  2405* 2447* 2458* 2459  2479  2481* 2513  2515* 2534  2963#
CK      012576  2732  2780#
CKDONE= 104007   632# 1157  1166  1177  1190  1207  1217  1232  1241  1252  1325  1373  1401
                1417  1421  1433  1440  1454  1475  1494  1512  1524  1538  1552  1567  1588
                1736  1979  2006
CLRCOR  006736  1971# 1973
CLRINT  011654   757  1075  1090  1109  1125  1140  1640  1687  1715  2044  2082  2128  2606#
CNTRLA  013306  2474  2873#
CNTRLC  013302   759  2872#
```

```
CORMAX  014024   741* 2021  2022  2024  2045  2060  2061  2094  2959#
CORSIZ  001434   707   712   719#
CORSZA  001522   733#
COUNT   014036  2964#
CRLF    013311  2163  2247  2494  2764  2875#
CYC     010034  2250# 2394
DATA    007644   767  2157#
DATA1   007662  2158  2161# 2187  2739
DATA2   007674  2165# 2172
DATA3   007726  2175# 2184
DECOCT= 104001   626#  703
DELAY1  013776  1069  1084  1103  1119  1134  1264  1296  1311  1344  1351  1358  1386  2587
                2948#
DELAY2  014000  2949#
DOT     013313   763  2877#
EMTOK   001220   647   649#
EMTSRV  001200   615   643#
EMTTAB  001240   651   657#
ERROR = 104400   624#  809   816   823   830   838   847   856   865   874   883   894   904
                 914   923   934   945   957   969   980   989   998  1009  1017  1022  1031
                1036  1044  1049  1072  1087  1093  1106  1124  1139  1149  1158  1167  1179
                1192  1208  1219  1234  1243  1253  1256  1270  1276  1282  1289  1303  1307
                1316  1326  1342  1349  1356  1363  1374  1393  1402  1409  1418  1422  1426
                1434  1438  1441  1444  1455  1466  1476  1487  1495  1501  1513  1525  1528
                1539  1542  1553  1556  1568  1571  1589  1597  1608  1647  1655  1665  1692
                1718  1737  1757  1779  1783  1787  1792  1817  1839  1902  1927  1951  1980
                1984  2007  2011  2014  2049  2087  2099  2132  2177  2181  2265  2274  2293
                2298  2306  2334  2339  2346  2376  2383  2390
ERR1    012504  2740  2758#
ERR2    012522  2759  2763#
ERTRAP  011704   607  2617#
EXTCHK  011612  2589  2593#
EXTTY   011244  2495#
GETREG= 104013   636#  1884  2475  2495  2540  2688  2724
GM03XB  014066  2976#
GM07XB  014064  2975#
HIGH    014060  2973#
ICOUNT  012704   799* 2795  2804#
INBUF   011264   765   768   771   774  2400* 2401  2446  2448* 2502# 2508
INCFLG  014020   728*  734*  1967  1995  2957#
INIT    001332   620   695#  2984
INITA   001532   726   735#
INITAD= 104011   634#  785   804   834   843   852   861   870   878   888   899   910   919
                 928   940   950   963   976   985   993  1004  1079  1098  1114  1129  1145
                1154  1164  1172  1185  1198  1213  1220  1225  1239  1249  1293  1322  1370
                1398  1414  1430  1450  1471  1492  1507  1518  1532  1546  1561  1577  1615
                1673  1701  1725  1743  1765  1798  1823  1861  1969  1997  2019  2058  2109
                2161  2254  2276  2314  2353  2785
INITB   001546   735   739#  761   2951
INIT1   001350   699#  717
INIT2   001624   746   761#  779
INIT3   001720   777#  2414  2952
INPUTA  011036  2449#  2450  2464  2480
INPUTB  011070  2457#  2490
KSTOR1  014040  2402* 2404* 2407* 2408* 2409* 2412  2415* 2418  2965#
KSTOR2  014042  2709* 2722* 2966#
```

```
KSTOR3  014044          2733* 2734* 2766  2771* 2773  2967#
KSTOR4  014046          2175* 2179* 2756  2968#
LDINTR  011620          1065  1080  1099  1115  1130  1624  1676  1706  1885  2029  2038  2075  2116
                        2257  2281  2323  2358  2597#
LODCOR  007042          2000# 2002
LOGERR  012344           617  2730#
LOGIC   001776           770   796#
LOW     014062          2974#
MASK    012342          2710* 2716  2721* 2726#
MEMSIZ  014022           730* 2958#
MESPRT  014056           793* 2758  2762* 2972#
MES10   013547          2743  2909#
MES11   013561          2746  2911#
MES12   013573          2749  2913#
MES13   013605          2752  2915#
MES2    013063           701  2843#
MES21   013617          2576  2918#
MES3    013101          2761  2846#
MES4    013172           745  2858#
MES40   013657          2621  2924#
MES41   013701          2626  2929#
MES42   013710          2393  2931#
MES43   013742          2243  2936#
MES44   013456          2245  2898#
MES5    013321           797  2881#
MES6    013340          2144  2884#
MES7    013361          2160  2887#
MES8    013510          2166  2903#
MES9    013530          2169  2906#
MONITR  001602           621   753#  2469  2577  2630
NOP   = 000240           344#
NULL  = 104010           633#   753   755
OCTPRT  012224           661  2704#
OUTPTA  011106          2461# 2462  2484  2488  2498
PC    =%000007           333#   757*  794*  798* 1058* 1065* 1075* 1080* 1090* 1099* 1109* 1115* 1125*
                        1130* 1140* 1229* 1509* 1520* 1534* 1548* 1563* 1579* 1617* 1624* 1640* 1675*
                        1676* 1687* 1703* 1706* 1715* 1734* 1751* 1773* 1811* 1833* 1869* 1885* 1976*
                        2004* 2029* 2038* 2044* 2075* 2082* 2115* 2116* 2128* 2157* 2167* 2170* 2194*
                        2249* 2256* 2257* 2281* 2323* 2326* 2358* 2367* 2399* 2442* 2602* 2610*
POP1SP= 005726           341#
POP2SP= 022626           343#  1074  1089  1108  1123  1138  1641  1686  1714  1893  1915  1943  2037
                        2043  2081  2127  2271  2295  2336  2380  2472  2618
PRINT = 104000           625#   697   700   715   744   758   762   777   796  2143  2159  2162  2165
                        2168  2243  2244  2247  2392  2473  2493  2499  2542  2575  2620  2625  2753
                        2760  2763
PRNTIT  012474          2743* 2746* 2749* 2752* 2754#
PROC    014032          2559* 2564  2962#
PRTADR  012556          2773#
PRTOCT= 104004           629#  2623  2628  2755  2765  2772
PSW     001270           671#   696*  790* 1067* 1082* 1101* 1117* 1132* 1908* 1938* 2563* 2601* 2606*
                        2683* 2706* 2799*
PUSH1S= 005746           340#
PUSH2S= 024646           342#
PVECTR  014006          2952#
PWRFAL  011470           609  2552#
PWRUP   011524          2560  2563#
```

```
AMARK   013316   716   778  2500  2543  2879#
RANBUF  014070  1569  1727  1745  1767  1777  1800  1825  1830  1837  1871  2319  2321  2360
                2363  2436  2979#
RESTRT  002002   798#  2145
RETURN  012710   789*  1059*  2250*  2417*  2800  2802*  2806#
R0     =X000000   326#  1384*  1389*  1580*  1583*  1593*  1599*  1604*  1610*  1620*  1622*  1629*  1630*
                1651*  1657*  1661*  1667*  1805*  1808*  1863*  1866*  1872*  1877*  1896*  1899*  1921*
                1924*  1931*  1933*  1945*  1946*  2024*  2026*  2045*  2047   2065*  2067*  2070*  2072*
                2083*  2085   2094*  2095*  2097   2164*  2171*  2173*  2183*  2261*  2262*  2285*  2291*
                2300*  2303*  2329*  2330*  2370*  2371*  2523*  2530   2552   2571*  2707*  2712*  2713*
                2714*  2715   2736   2770*  2771   2779#
R1     =X000001   327#   704*   709*   719    724*   725*  729*  730   736*  737   739   740*  741
                 954*   955*  1069*  1070*  1084*  1085*  1103*  1104*  1119*  1120*  1134*  1135*  1264*
                1265*  1294*  1299*  1461*  1483*  1484*  1485   1497*  1498*  1499   1594*  1595   1605*
                1606   1619*  1621*  1652*  1653   1662*  1663   1726*  1728   1729*  1730   1744*  1746
                1747*  1748   1766*  1768   1769*  1770   1799*  1801   1802*  1803   1806*  1807*  1824*
                1826   1827*  1828   1830*  1831*  1864*  1865*  1871*  1876*  1883*  1895*  1897   1920*
                1922   1930*  1932*  1948*  1949   1970*  1971*  1972   1981*  1982   1999*  2000*  2001
                2008*  2009   2025*  2027*  2046*  2051*  2066*  2068*  2071*  2073*  2084*  2089*  2096*
                2100*  2110*  2111*  2112   2129*  2130   2134   2174*  2178*  2182*  2277*  2278*  2279
                2288*  2289*  2299*  2301   2315*  2316*  2317   2319*  2320*  2321   2328*  2332*  2340*
                2341   2343   2354*  2355*  2356   2360*  2361*  2362*  2363   2365*  2369*  2373*  2384*
                2385   2387   2401*  2403*  2404   2451*  2452*  2453   2455   2457   2463   2467   2470
                2477   2482*  2485   2489   2491   2497   2511*  2524   2525*  2526*  2527*  2528*  2529*
                2530*  2533   2553   2570*  2572*  2573*  2587*  2590*  2645   2668*  2737   2741   2744
                2747   2750   2769*  2770   2778*
R2     =X000002   328#   705*   710*   731*   732   1250*  1258*  1296*  1297*  1311*  1312*  1323*  1328*
                1344*  1345*  1351*  1352*  1358*  1359*  1371*  1376*  1386*  1387*  1453*  1457*  1462*
                1463*  1464   1474*  1478*  1581*  1582*  1727*  1728*  1745*  1746*  1767*  1768*  1800*
                1801*  1825*  1826*  2284*  2512*  2524*  2528   2529   2554   2569*  2646   2667*  2680*
                2686   2690   2692   2698   2715*  2716*  2717*  2720   2738   2777*
R3     =X000003   329#   708*   713*  1873*  1874   1879*  1880   2509*  2533*  2555   2568*  2647   2666*
R4     =X000004   330#  1874*  1875*  1876   2446*  2457*  2483   2508*  2516   2518   2520   2522*  2523
                2532   2556   2567*  2648   2665*
R5     =X000005   331#  2557   2566*  2649   2664*
SAVEPC  014010  2641*  2653   2660*  2672   2953#
SAVPSW  014012  2642*  2652   2661*  2671   2954#
SAVREG= 104012   635#  1862   2445   2507   2679   2704
SAV2PC  014014  2643*  2651   2662*  2670   2955#
SAV2SW  014016  2644*  2650   2663*  2669   2956#
SCOPE  = 104002   627#   806    813    820    827    833   842   851   860   869   877   887   898
                 909    918    927    939    949    962   975   984   992  1003  1012  1026  1039
                1056   1078   1097   1113   1128   1144  1153  1163  1171  1184  1197  1212  1224
                1238   1248   1292   1321   1369   1397  1413  1429  1449  1470  1491  1506  1517
                1531   1545   1560   1576   1614   1672  1700  1724  1742  1764  1797  1822  1860
                1966   1994   2018   2057   2108   2141  2253  2275  2313  2352  2391
SCOPEB  012660  2792   2798#
SCOPEC  012622   659   2790#
SCOPEF  012706   791*  2416*  2795   2797*  2801*  2805#
SCOPEG  012672  2794   2796   2801#
SETREG  010756  1058   1229   1509   1520   1534   1548  1563  1579  1617  1675  1703  1734  1751
                1773   1811   1833   1869   1976   2004  2115  2256  2284  2326  2367  2436#
SETUP   001726   785#   798    2157   2249   2399
SHIFT   012266  2712#  2723
SIGNBF  013774   719*   720*   721*  2947#
SIZE    001410   709#   714
```

```
SOFLAG  014026          792*  2960#
SP    =X000006          332#   643*   644*   645*   646   649*   650*   651*   652*   653   695*   756*  2242*
                       2552*  2553*  2554*  2555*  2556*  2557*  2558*  2559   2564*  2565  2566   2567  2568
                       2569   2570   2571   2593*  2597   2598*  2617   2619   2641   2642  2643   2644  2645*
                       2646*  2647*  2648*  2649*  2650*  2651*  2652*  2653*  2660   2661  2662   2663  2664
                       2665   2666   2667   2668   2669*  2670*  2671*  2672*  2680   2681* 2707   2708* 2733
                       2736*  2737*  2738*  2777   2778   2779   2798   2802
SPACE = 104003          628#  2767   2774
SPACEX  010754         2428*  2430*  2432#  2819*  2823*
SPCHR1  011122         2454   2456   2467#
SPCHR2  011206         2478   2485#
SPCHR3  011224         2486   2489#
SPCHR4  011232         2491#
SPCHR5  011250         2460   2492   2497#
STACK   014002          695    756   2242   2950#
SWR     001302          676#  2731   2780   2783   2791   2793
SWR0    001304          677#
SW00  = 000001          322#
SW01  = 000002          321#
SW02  = 000004          320#
SW03  = 000010          319#
SW04  = 000020          318#
SW05  = 000040          317#
SW06  = 000100          316#
SW07  = 000200          315#
SW08  = 000400          314#
SW09  = 001000          313#
SW10  = 002000          312#
SW11  = 004000          311#
SW12  = 010000          310#  2783
SW13  = 020000          309#
SW14  = 040000          308#
SW15  = 100000          307#
SYNC    010042         2250   2253#
SYNC0   010004          773   2242#
SYNC1   010114         2258   2266   2271#
SYNC2   010256         2294   2305   2313#
SYNC3   010420         2335   2345   2352#
SYNRET  006664         1935   1943#
SYNRE1  010220         2282   2285#
SYNRE2  010364         2324   2336#
SYNRE3  010542         2359   2377   2390#
TAGAA   003106         1066   1074#
TAGAAA  003154         1081   1089#
TAGAAB  007236         2039   2042#
TAGAAR  007260         2047#  2052
TAGAB   003230         1100   1108#
TAGAC   003274         1116   1123#
TAGAD   003342         1131   1138#
TAGAF   005214         1582#  1584
TAGAG   005254         1595#  1600
TAGAH   005302         1606#  1611
TAGAI   005354         1621#  1623
TAGAJ   005424         1625   1637#
TAGAK   005512         1663#  1668
TAGAL   005666         1707   1714#
```

```
TAGAM   007012          1968    1983#
TAGAN   006046          1768#   1771
TAGAO   007220          2030    2036#
TAGAP   007276          2053#
TAGAQ   007424          2076    2080#
TAGAR   007446          2085#   2090
TAGAS   007500          2097#   2101
TAGF    003700          1251#   1259
TAGG    003730          1263#
TAGL    004526          1454#   1460
TAGM    004544          1458    1461#
TAGO    004604          1475#   1481
TAGP    004622          1479    1483#
TAGRAB  006562          1905    1915#
TAGRAN  006500          1886    1893#
TAGXF   004014          1295#   1300
TAGXJ   005464          1653#   1658
TAGXK   004250          1372#   1377
TAGXL   004106          1324#   1329
TAGXM   004312          1385#   1390
TAGXN   005766          1746#   1749
TAGYA   006164          1801#   1804
TAGYB   006206          1807#   1809
TAGYC   006266          1826#   1829
TAGYF   007120          1996    2015#
TAGYK   005570          1677    1685#
TAGYN   005716          1728#   1731
TAGIB   007574          2117    2126#
TAGIC   007612          2130#   2135
TEMP1   014050          2617*   2622*   2624    2768*   2775*   2969#
TEMP2   014052          2619*   2627*   2629    2970#
TEMP3   014054          2971#
TESTX   010610           776    2399#   2412
TITLE   012755           698    2829#
TKB     001274           673#   2451
TKS     001272           672#   2449    2682*   2705*   2810
TKSFLG  012712           663    2810#
TPB     001300           675#   2427*   2463*   2692*   2694*   2697*   2720*   2822*
TPS     001276           674#   2185    2425    2461    2684    2695    2718    2820
TSTA    010630          2403#   2410
TSTB    010664          2406    2412#
TSTTKS= 104006           631#   2730    2790
TSTXB   003044          1059    1062#
TST1    002014           789     803#   2806
TST10   002136           851#
TST100  006252          1822#
TST101  006344          1860#
TST102  006720          1966#
TST103  007016          1994#
TST104  007122          2018#
TST105  007300          2057#
TST106  007514          2108#
TST107  007632          2141#
TST11   002162           860#
TST12   002206           869#
TST13   002226           877#
```

```
TST14   002252            887#
TST15   002276            898#
TST16   002316            909#
TST17   002336            918#
TST2    002020            806#
TST20   002356            927#
TST21   002410            939#
TST22   002436            949#
TST23   002472            962#
TST24   002524            975#
TST25   002542            984#
TST26   002560            992#
TST27   002604           1003#
TST3    002030            813#
TST30   002632           1012#
TST31   002702           1026#
TST32   002752           1039#
TST33   003022           1056#
TST34   003114           1078#
TST35   003170           1097#
TST36   003236           1113#
TST37   003304           1128#
TST4    002042            820#
TST40   003352           1144#
TST41   003372           1153#
TST42   003414           1163#
TST43   003430           1171#
TST44   003456           1184#
TST45   003512           1197#
TST46   003554           1212#
TST47   003604           1224#
TST5    002054            827#
TST50   003650           1238#
TST51   003670           1246#
TST52   004004           1292#
TST53   004076           1321#
TST54   004240           1369#
TST55   004340           1397#
TST56   004372           1413#
TST57   004434           1429#
TST6    002066            833#
TST60   004506           1449#
TST61   004564           1470#
TST62   004640           1491#
TST63   004674           1506#
TST64   004726           1517#
TST65   004776           1531#
TST66   005046           1545#
TST67   005116           1560#
TST7    002112            842#
TST70   005166           1576#
TST71   005320           1614#
TST72   005530           1672#
TST73   005616           1700#
TST74   005702           1724#
TST75   005752           1742#
```

```
TST76   006032          1764#
TST77   006150          1797#
TTYIN = 104005           630#   702    764    2189   2246   2544   2812
TYPECL  012176          2691    2694#
TYPERA  012142          2684#   2685   2693   2699
TYPERI  012162          2687    2690#
TYPER2  012170          2692#
TYPNES  012116           657    2679#
WATINP  007766          2167    2170   2189#
X     = 000110           345#    806    807#   813    814#   820    821#   827    828#   833    834#   842    843#
                         851     852#   860    861#   869    870#   877    878#   887    888#   898    899#   909
                         910#    918    919#   927    928#   939    940#   949    950#   962    963#   975    976#
                         984     985#   992    993#  1003   1004#  1012   1013#  1026   1027#  1039   1040#  1056
                        1057#   1078   1079#  1097   1098#  1113   1114#  1128   1129#  1144   1145#  1153   1154#
                        1163    1164#  1171   1172#  1184   1185#  1197   1198#  1212   1213#  1224   1225#  1238
                        1239#   1248   1249#  1292   1293#  1321   1322#  1369   1370#  1397   1398#  1413   1414#
                        1429    1430#  1449   1450#  1470   1471#  1491   1492#  1506   1507#  1517   1518#  1531
                        1532#   1545   1546#  1560   1561#  1576   1577#  1614   1615#  1672   1673#  1700   1701#
                        1724    1725#  1742   1743#  1764   1765#  1797   1798#  1822   1823#  1860   1861#  1966
                        1967#   1994   1995#  2018   2019#  2057   2058#  2108   2109#  2141   2142#
XCHK1   011600          2588#   2591
XCKDON  011574           664    2587#
XGETRG  012042           668    2660#
XINIT   011756           666    2635#
XLOOP   010706          2416#   2417
XNULL   012724           665    2819#  2824
XPRT    012546          2770#   2776
XPRT1   012570          2757    2777#
XSAVRG  011766           667    2641#
XSPACE  010724           660    2425#  2429
XTTYIN  011020           612     662   2445#  2501
.     = 015074           348#    349    351    353    355    357    359    361    363    365    367    369    371
                         373     375    377    379    381    383    385    387    389    391    393    395    397
                         399     401    403    405    407    409    411    413    415    417    419    421    423
                         425     427    429    431    433    435    437    439    441    443    445    447    449
                         451     453    455    457    459    461    463    465    467    469    471    473    475
                         477     479    481    483    485    487    489    491    493    495    497    499    501
                         503     505    507    509    511    513    515    517    519    521    523    525    527
                         529     531    533    535    537    539    541    543    545    547    549    551    553
                         555     557    559    561    563    565    567    569    571    573    575    577    579
                         581     583    585    587    589    591    593    595    597    599    601    603    606#
                         611#    614#   619#   641#   723    738    766    769    772    775    808    815    822
                         829     837    846    855    864    873    882    893    903    913    922    933    944
                         956     968    979    988    997   1008   1016   1021   1030   1035   1043   1048   1071
                        1073    1086   1088   1092   1105   1107   1121   1122   1136   1137   1148   1175   1178
                        1191    1218   1233   1242   1255   1266   1269   1275   1281   1288   1298   1302   1306
                        1313    1315   1341   1346   1348   1353   1355   1360   1362   1388   1392   1408   1425
                        1437    1443   1465   1486   1500   1527   1541   1555   1570   1596   1598   1607   1609
                        1631    1639   1646   1654   1656   1664   1666   1689   1691   1717   1754   1756   1776
                        1778    1782   1786   1791   1814   1816   1836   1838   1947   1983   2010   2013   2028
                        2048    2050   2069   2074   2086   2088   2098   2113   2131   2133   2176   2180   2186
                        2192    2273   2290   2297   2338   2382   2413   2426   2468   2471   2503#  2574   2696
                        2711    2719   2742   2745   2748   2751   2781   2784   2811   2821   2942#  2980#
```

```
COMMEN          18#
ENDCOM          18#
ESCAPE          18#
GETPRI          18#
GETSWR          18#
MULT            18#
NEWTST          18#
POP             18#
PUSH            18#
REPORT          18#
SETPRI          18#
SETUP           18#
SKIP            18#
SLASH           18#
STARS           18#
SWRSU           18#
TESTN         622#    806    813    820    827    833    842    851    860    869    877    887    898    909    918
               927    939    949    962    975    984    992   1003   1012   1026   1039   1056   1078   1097   1113
              1128   1144   1153   1163   1171   1184   1197   1212   1224   1238   1248   1292   1321   1369   1397
              1413   1429   1449   1470   1491   1506   1517   1531   1545   1560   1576   1614   1672   1700   1724
              1742   1764   1797   1822   1860   1966   1994   2018   2057   2108   2141
TS            622#    806    813    820    827    833    842    851    860    869    877    887    898    909    918
               927    939    949    962    975    984    992   1003   1012   1026   1039   1056   1078   1097   1113
              1128   1144   1153   1163   1171   1184   1197   1212   1224   1238   1248   1292   1321   1369   1397
              1413   1429   1449   1470   1491   1506   1517   1531   1545   1560   1576   1614   1672   1700   1724
              1742   1764   1797   1822   1860   1966   1994   2018   2057   2108   2141
TYPBIN          18#
TYPDEC          18#
TYPNAM          18#
TYPNUM          18#
TYPOCS          18#
TYPOCT          18#
TYPTXT          18#
$$ESCA          18#
$$NEWT          18#
$$SKIP          18#
.EQUAT          18#
.HEADE          18#
.KT11           18#
.SETUP          18#
.SWRHI          18#
.SACTI          18#
.SAPTB          18#
.SAPTH          18#
.SAPTY          18#
.SASTA          18#
.SCATC          18#
.SCNTA          18#
.SD820          18#
.SD820          18#
.SDIV           18#
.SEOP           18#
.SERRO          18#
.SERRT          18#
.SMULT          18#
.SPOWE          18#
```

C07


```
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$4DCA      1#
.1170       1#
```

```
ADD     651  729 1618 1870 1875 2023 2062 2064 2095 2404 2415 2441 2528 2529 2530
       2593 2598 2681 2708
ASL     649  709  710  724  731 2407 2408 2409 2440 2525 2526 2527
ASR     720
BEQ     707  712  723  808  815  822  829  837  846  855  864  873  893  922  933
        944  956  968 1008 1016 1021 1030 1035 1043 1048 1255 1288 1306 1341 1348
       1355 1362 1392 1408 1425 1437 1458 1465 1479 1486 1500 1527 1541 1555 1570
       1607 1646 1654 1689 1691 1756 1778 1782 1786 1791 1816 1838 1878 1898 1950
       1968 1983 1996 2010 2098 2302 2342 2386 2406 2480 2490 2498 2517 2691 2784
BGT    2429 2521
BIC     650  931  942  953  955 1188 1201 1205 1215 1227 1257 1327 1375 1463 1484
       1498 1919 1944 2264 2403 2452 2522 2607 2735
BICB   2716
BIS     725  900  911  929  941  951  952  964  965  966  977  986  994  995 1005
       1013 1018 1019 1027 1032 1033 1040 1045 1046 1064 1068 1083 1101 1118 1133
       1146 1186 1187 1199 1200 1214 1226 1286 1637 1685 1704 1831 1883 1907 1937
       2036 2042 2080 2126 2193 2365 2375 2600 2635 2717
BIT     881  892  912 1147 1254 1268 1287 1301 1305 1407 1436 1442 2012 2731 2783
       2791 2793
BITB   2718
BLE    2514
BLT    2519
BMI     979  988  997 1281 1315 1717 2180 2454 2456 2460 2589
BNE     647  766  769  772  775  882  913 1071 1086 1105 1121 1136 1148 1259 1266
       1269 1298 1300 1302 1313 1329 1346 1353 1360 1377 1388 1390 1443 1584 1596
       1600 1611 1623 1631 1658 1664 1668 1731 1749 1771 1804 1809 1829 1867 1881
       1900 1923 1925 1934 1947 1973 2002 2013 2028 2048 2052 2069 2074 2086 2090
       2101 2113 2131 2135 2172 2184 2263 2280 2290 2292 2304 2318 2322 2331 2333
       2344 2357 2364 2372 2374 2388 2468 2471 2478 2486 2492 2535 2574 2591 2687
       2723 2732 2740 2742 2745 2748 2751 2759 2776 2792 2794 2824
BPL     714  903 1092 1175 1275 1639 1754 1776 1814 1836 2176 2186 2192 2273 2297
       2338 2382 2413 2426 2450 2462 2537 2539 2685 2696 2719 2781 2796 2811 2821
BR      717  738  746  779 1073 1088 1107 1122 1137 1178 1191 1218 1233 1242 1460
       1481 1598 1609 1656 1666 1882 1901 1926 2050 2088 2133 2187 2266 2294 2305
       2335 2345 2377 2389 2410 2464 2484 2488 2501 2531 2693 2699 2711 2757
CLR     699  728  736  791  792  793 1006 1014 1028 1041 1324 1372 1380 1382 1873
       1908 1938 1971 2174 2190 2261 2288 2329 2370 2400 2402 2416 2430 2447 2448
       2510 2511 2512 2572 2601 2609 2683 2706 2801 2822
CMP     774  836  845  854  863  872  921  932  943  967 1020 1034 1047 1340 1347
       1361 1391 1424 1464 1485 1499 1526 1540 1554 1569 1595 1606 1645 1653 1663
       1690 1730 1748 1755 1770 1777 1781 1785 1789 1803 1815 1828 1837 1880 1897
       1922 1949 1972 1982 2001 2047 2085 2097 2112 2130 2134 2279 2301 2317 2321
       2341 2343 2356 2363 2385 2387 2412 2459 2520 2536 2538 2739 2744 2747 2750
       2795 2798
CMPB    765  768  771 2453 2455 2467 2470 2477 2485 2489 2491 2497 2516 2518 2686
       2690
COM     721
DEC     711  713 1120 1135 1258 1299 1328 1345 1352 1359 1376 1387 1389 1457 1478
       1667 2027 2051 2068 2073 2089 2100 2171 2183 2405 2428 2481 2515 2722 2775
       2823
EMT     625  626  627  628  629  630  631  632  633  634  635  636
HALT    648 2561 2782
INC     734  890 1070 1085 1104 1165 1173 1231 1265 1297 1312 1420 1439 1459 1480
       1583 1599 1610 1622 1627 1630 1657 1679 1729 1747 1769 1802 1808 1827 1866
       1877 1879 1899 1924 1933 1946 2032 2178 2182 2262 2289 2291 2303 2330 2332
       2371 2373 2458 2573 2590 2762 2797
```

```
JMP      620  621  653  767  770  773  776 2145 2394 2414 2418 2469 2476 2545 2577
        2630 2800
JSR      757  798 1058 1065 1075 1080 1090 1099 1109 1115 1125 1130 1140 1229 1509
        1520 1534 1548 1563 1579 1617 1624 1640 1675 1676 1687 1703 1706 1715 1734
        1751 1773 1811 1833 1869 1885 1976 2004 2029 2038 2044 2075 2082 2115 2116
        2128 2157 2167 2170 2249 2256 2257 2281 2284 2323 2326 2358 2367 2399
MOV      643  645  652  695  696  704  705  708  719  726  727  730  732  735  741
         742  743  756  761  786  787  788  789  790  799  835  844  853  862  871
         879  880  889  920  930  954 1057 1059 1067 1069 1082 1084 1102 1103 1117
        1119 1132 1134 1155 1156 1189 1202 1216 1228 1230 1240 1250 1251 1263 1264
        1294 1295 1296 1311 1323 1333 1334 1335 1336 1344 1351 1358 1371 1378 1379
        1381 1384 1386 1399 1400 1415 1416 1431 1432 1451 1452 1453 1461 1462 1472
        1473 1474 1483 1493 1497 1508 1510 1511 1519 1521 1522 1523 1533 1535 1536
        1537 1547 1549 1550 1551 1562 1564 1565 1566 1578 1580 1581 1582 1585 1586
        1587 1593 1594 1604 1605 1616 1619 1620 1621 1626 1628 1629 1632 1651 1652
        1661 1662 1674 1678 1680 1702 1705 1708 1709 1726 1727 1728 1733 1735 1744
        1745 1746 1750 1752 1766 1767 1768 1772 1774 1799 1800 1801 1805 1806 1807
        1810 1812 1824 1825 1826 1830 1832 1834 1863 1864 1865 1868 1871 1872 1874
        1876 1887 1888 1894 1895 1896 1905 1906 1920 1921 1930 1931 1932 1935 1936
        1945 1948 1970 1974 1975 1977 1978 1981 1998 1999 2000 2003 2005 2008 2020
        2021 2022 2024 2025 2026 2031 2033 2045 2046 2059 2060 2061 2063 2065 2066
        2067 2070 2071 2072 2077 2083 2084 2094 2096 2110 2111 2114 2118 2119 2120
        2129 2158 2164 2173 2175 2179 2242 2250 2255 2259 2260 2277 2278 2283 2285
        2286 2287 2299 2300 2315 2316 2319 2320 2325 2327 2328 2340 2354 2355 2360
        2361 2362 2366 2368 2369 2384 2401 2417 2427 2436 2437 2438 2446 2451 2457
        2482 2487 2508 2509 2523 2524 2533 2552 2553 2554 2555 2556 2557 2558 2559
        2560 2563 2564 2565 2566 2567 2568 2569 2570 2571 2587 2597 2599 2606 2608
        2617 2619 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653
        2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2680 2682
        2694 2697 2705 2707 2709 2710 2721 2733 2736 2737 2738 2743 2746 2749 2752
        2768 2769 2770 2771 2777 2778 2779 2799 2802 2819
MOVB     2463 2692 2715 2720
NEG      2439
MAP       803 1062 1203 1204 1952 2015 2053
RESET     754 2248
ROL      2712 2713 2714
RTI      2431 2496 2541 2592 2594 2636 2654 2673 2689 2725 2786 2803 2813 2825
RTS       794 2194 2442 2602 2610
SUB       644  706  740 2622 2627 2734
TRAP      624
TST       646  722  733  737  739  807  814  821  828  891  901  935  946  958  970
          978  987  996  999 1007 1015 1029 1042 1159 1168 1176 1180 1193 1209 1235
         1244 1280 1304 1317 1343 1350 1354 1357 1385 1403 1419 1423 1435 1445 1456
         1477 1496 1716 1967 1995 2009 2272 2296 2337 2381 2479 2483 2513 2532 2534
         2741 2758 2780
TSTB      902 1091 1174 1274 1314 1638 1688 1753 1775 1813 1835 2185 2191 2425 2449
         2461 2588 2684 2695 2698 2810 2820
WAIT     1633 1681 1710 1889 1909 1939 2034 2040 2078 2121
.ABS      298
.ASCII   2829 2838 2843 2846 2854 2858 2865 2872 2873 2875 2877 2879 2881 2884 2887
         2891 2898 2903 2906 2909 2911 2913 2915 2918 2924 2929 2931 2936
.BYTE    2828
.ENABL      1
.END     2984
.EVEN    2942
.LIST       1  622  807  814  821  828  834  843  852  861  870  878  888  899  910
```

```
                919    928    940    950    963    976    985    993   1004   1013   1027   1040   1057   1079   1098
               1114   1129   1145   1154   1164   1172   1185   1198   1213   1225   1239   1249   1293   1322   1370
               1398   1414   1430   1450   1471   1492   1507   1518   1532   1546   1561   1577   1615   1673   1701
               1725   1743   1765   1798   1823   1861   1967   1995   2019   2058   2109   2142
.MACRO            1    622
.NLIST            1    622    807    814    821    828    834    843    852    861    870    878    888    899    910
                919    928    940    950    963    976    985    993   1004   1013   1027   1040   1057   1079   1098
               1114   1129   1145   1154   1164   1172   1185   1198   1213   1225   1239   1249   1293   1322   1370
               1398   1414   1430   1450   1471   1492   1507   1518   1532   1546   1561   1577   1615   1673   1701
               1725   1743   1765   1798   1823   1861   1967   1995   2019   2058   2109   2142
.REM              1
.REPT           349
.TITLE          297


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZADGA.SEQ=SYSMAC.CO,DZADGA.CMB
RUN-TIME:  27 37  4 SECONDS
RUN-TIME RATIO:  128/70=1.8
CORE USED:  33K  (65 PAGES)
```

G07

Spooler runtime 10 Seconds, 46 KCS, 282 disk reads, 4 disk writes, 83 pages

```
 10                      ...B1           007644  004767  ...B5
 35                      ...C1                           ...C5
 89                      ...D1           010100  005200  ...D5
                         ...E1           010276  001373  ...E5
148                      ...F1           010524  001376  ...F5
                         ...G1           010640  005367  ...G5
206                      ...H1           011054  120127  ...H5
                         ...I1           011304  104012  ...I5
266                      ...J1           011514  012767  ...J5
322             000001   ...K1           011662  042777  ...K5
355  000014     000016   ...L1           012012  010346  ...L5
411  000174     000176   ...M1           012146  100375  ...M5
467  000354     000356   ...N1           012400  010146  ...N5

523  000534     000536   ...B2           012620  000002  ...B6
579  000714     000716   ...C2           013026  045055  ...C6
                104012   ...D2           013405     123  ...D6
     001216     000000   ...E2                           ...E6
     001354     104000   ...F2           014012  000000  ...F6
     001610     016706   ...G2                           ...G6
     001760     005067   ...H2  DATA3    007726          ...H6
     002114     104011   ...I2  MEMSIZ   014022          ...I6
     002252     104002   ...J2                           ...J6
     002406     005713   ...K2  SPCHR1   011122          ...K6
     002466     104400   ...L2  TAGL     004526          ...L6
                         ...M2  TST24    002524          ...M6
     003010     022777   ...N2  X      = 000110          ...N6

     003146     001376   ...B3  REPORT       1#  CROSS RE ...B7
     003336     001376   ...C3  .$SIZE       1#  CROSS RE ...C7
     003452     104400   ...D3  BGT        2429     2521  ...D7
     003602     104011   ...E3             1119     1132  ...E7
     003716     104400   ...F3             1725     1743  ...F7
     004032     001370   ...G3   **END** USER DAVIES,TOM  ...G7
     004166     022714   ...H3
     004334     001401   ...I3
     004462     001401   ...J3
     004570     012714   ...K3
     004722     104007   ...L3
     005102     104400   ...M3
                         ...N3

     005360     005200   ...B4
                         ...C4
     005702     104002   ...D4
     005772     022701   ...E4
     006170     022701   ...F4
                         ...G4
     006542     012777   ...H4
                         ...I4
                         ...J4
     007020     005767   ...K4
     007166     012720   ...L4
     007350     017700   ...M4
     007530     022701   ...N4
```