.REM !

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DDDRA-A-D
PRODUCT NAME:	DR11C DEVICE REGISTER TEST
DATE RELEASED:	21 DECEMBER 1975
MAINTAINER:	DIAGNOSTIC GROUP

1. ABSTRACT

THIS IS A LOGIC TEST OF THE DR11C. FOR THIS TEST TO OPERATE
A SPECIAL MAINTENANCE CABLE MUST BE CONNECTED (BCO8R).
THIS TEST WILL CHECK UP TO 32 SEQUENTIAL DR11C'S.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 STANDARD COMPUTER

DR11C

BCO8R FOR EACH DR11C

2.2 STORAGE

2.2.1 THE PROGRAM LOADS INTO 4K OF MEMORY WITH BOOTSTRAP

3. LOADING PROCEDURE

3.1 METHOD

ABSOLUTE LOADER

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTING

STARTING AT SA 200 ALL SWITCHES SHOULD BE DOWN OR ZERO.
(IF NOT ZERO, BIT 0 TO 8 WILL BE STARTING VECTOR.)

4.2 STARTING ADDRESS OR ADDRESSES

(A) 200 = START OF TEST--FOR NORMAL TESTING
(B) 204 = SPECIAL ENTRANCE --FOR TESTING UNIQUE DR11C
(C) 210 = RESTART--FOR STARTING AFTER SHUT DOWN

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY.
IF THERE IS NO TTY IN THE SYSTEM,
PATCH THE TAGS $TPS AND $TPB WITH
177570
SET SWITCH REGISTER TO STARTING ADDRESS.
LOAD ADDRESS.
PRESS START.
THE PROGRAM WILL STAY IN SECTION AND LOOP.

4.3.1 FOR SPECIAL ENTRANCE - SA204

```
1ST HALT SET SWITCH REGISTER EQUAL TO CSR ADDRESS OF DR11C
PRESS CONTINUE
2ND HALT SET SWITCH REGISTER EQUAL TO VECTOR ADDRESS OF DR11C
PRESS CONTINUE
RAISE SWITCH 10 TO "1" TO INHIBIT SEQUENCING TO NEXT DR11C
```

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 AT SA 200 .. THE INSTRUCTION AND LOGIC TEST.
WITH ALL SWITCHES DOWN THE PROGRAM WILL PRINT
OUT ON ERRORS AND CONTINUE IN TEST.
("/ DEVICE ADDRESS VECTOR ADDRESS" WILL
BE PRINTED AT COMPLETION OF TESTING EACH DR11C IF
SW06 IS DOWN. IF SW06 IS UP THEN NOTHING WILL BE
PRINTED.
INSTEAD THERE WILL BE "5" ON THE DISPLAY LIGHTS
FOR COUPLE OF SECONDS AND THEN PROGRAM WILL
HALT AT LOCATION 252.
PRESS CONTINUE TO RESTART.)

5.1.2 SWITCH SETTINGS ARE

SW15 = 1 OR UP ... HALT ON ERROR
SW14 = 1 OR UP ... SCOPE LOOP
SW13 = 1 OR UP ... INHIBIT PRINTOUT
SW12 = 1 OR UP ... NOT USED
SW11 = 1 OR UP ... INHIBIT ITERATION LOOP
SW10 = 1 OR UP ... DO NOT ADVANCE TO NEXT DR11C
SW09 = 1 OR UP ... INHIBIT PRINTOUT OF DEVICE TESTED.
SW06 = 1 OR UP ...HALT ON END OF PASS.

5.2. SUBROUTINE ABSTRACTS

5.2.1 BEGIN SA 200

5.2.2 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST
IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING
ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED.
IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE
START OF THE SUBTEST THAT THE SCOPE LOOP IS RE-
QUESTED FOR.

5.2.3 HALT

IS A ROUTINE THAT PRINTS-OUT AN ADDRESS THAT TAGS
THE FAILING SUBTEST, AND THE INCORRECT DATA AT
THE TIME OF THE FAILURE, IF THERE IS A TTY.
IT ALSO STORES AWAY THE PC+2 OF THE
DETECTED ERROR AND THE PROCESSOR STATUS REGISTER
STARTING AT LOCATION 017400.

.5. OPERATING PROCEDURE CONT'D)

5.3 PROGRAM AND/OR OPERATOR ACTION

EITHER A) ONLY IF THERE IS A TTY ---

5.3.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

5.3.2 PATCH THE TAGS $TPS AND $TPB
WITH 177570

5.3.3 LOAD ADDRESS 000200

5.3.4 CLEAR CONSOLE SWITCHES

5.3.5 SET SW15=1,SW11=1,SW07=1,SW06=1

5.3.6 PRESS START

IF THERE IS NO ERROR, PROGRAM WILL RUN
FOR A FEW SECONDS, AT THE END OF WHICH THERE WILL BE "5"
ON THE DISPLAY LIGHTS FOR COUPLE OF SECONDS AND THEN
PROGRAM WILL HALT AT ADDRESS 252 FOR END OF PASS.
PRESS CONTINUE TO CONTINUE THE PROGRAM.

IF ANY ERROR ENCOUNTERED, PROGRAM WILL HALT AT ADDRESS
3746.

TO DETERMINE TYPE OF ERROR:

5.3.7 LOAD ADDRESS 017400

5.3.8 PRESS EXAMINE

CONTENT OF THIS LOCATION IS ---PC+2 OF THE DETECTED ERROR

5.3.9 PRESS EXAMINE

CONTENT IS --- PROCESSOR STATUS REGISTER

TO TEST IF THERE IS ANY MORE ERROR IN ANY OTHERE TEST:

5.3.10 LOAD ADDRESS 000200

5.3.11 CLEAR CONSOLE SWITCHES

5.3.12 SET SW15=1,SW11=1,SW07=1,SW06=1

5.3.13 PRESS START

PROGRAM WILL HALT FOR THE FIRST ERROR THAT WAS JUST OBSERVED

5.3.14 PRESS CONTINUE

IF THE PROGRAM HALTS AT ADDRESS 3746:

5.3.15 LOAD ADDRESS 017400

5.3.16 PRESS EXAMINE

CONTENT IS THE NEW PC+2 OF THE DETECTED ERROR

NEXT LOCATION CONTAINS PROCESSOR STATUS REGISTER

STEPS 5.3.10 THROUGH 5.3.16 SHOULD BE REPEATED

FOR ANY FURTHER ERROR.

OR B) IF THERE IS A TTY ---

5.3.1 LOAD PROGRAM INTO MEMORY

5.3.2 DO NOT PATCH ANY WHERE

5.3.3 CLEAR CONSOLE SWITCHES

5.3.4 PRESS START

PROGRAM WILL TYPE THE DEVICE ADDRESS FOLLOWED BY THE VECTOR

ADDRESS.

THEN IF THERE IS NO ERROR AND IF SW06 IS DOWN,

"/ DEVICE REGISTER VECTOR ADDRESS" WILL BE

TYPED OUT FOR END OF PASS. THERE WILL BE "5"

ON THE DISPLAY LIGHTS BEFORE THE TYPE OUT.

IF SW06 IS UP PROGRAM WILL HALT AT 252 AND

THERE WILL BE NO TYPE OUT.

IF THERE IS ANY ERROR PROGRAM WILL TYPE OUT:

PC+2 OF DETECTED ERROR PROCESSOR STATUS REGISTER

DEVICE ADDRESS VECTOR ADDRESS. (ALL FOUR IN ONE LINE)

IF SW15 IS UP PROGRAM WILL HALT AT 3746 AFTER TYPE OUT.

PRESS CONTINUE TO CONTINUE THE PROGRAM.

6. ERRORS

6.1 ERROR PRINTOUT

ARE IN A FOUR WORD FORMAT. THE 1ST IS THE PC+2 OF THE DETECTED ERROR. THE 2ND IS THE PROCESSOR STATUS REGISTER. THE 3RD IS DEVICE ADDRESS. THE 4TH IS VECTOR ADDRESS.

6.2 ERROR RECOVERY

DEPRESS CONTINUE TO RESTART SECTION

7. RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

THE DR11C MUST HAVE THE 8CO8R CABLE TO RUN
THIS TEST.

NOTE THAT THE DR11C HAS FLOATING VECTORS:

THE BELOW IS THE ASSIGNMENT OF FLOATING VECTORS, THE ASSIGNED
SEQUENCES ARE:

1. STARTING AT 300 AND WORKING UPWARD ALL DC11'S WILL BE ASSIGNED.

2. THEN ANY EXTRA KL11 CALLED FOR (VT05, VT06, LC11)

3. THEN ANY DP11 CALLED FOR.

4. THEN ANY DM11 CALLED FOR.

5. THEN ANY DN11 CALLED FOR.

6. THEN ANY DM11BB CALLED FOR.

7. THEN ANY DR11A CALLED FOR.

8. THEN ANY DR11C CALLED FOR.

THE DR11A AND DR11C DEVICE ADDRESSES WILL BE ASSIGNED IN THE USER
AREA OF 767776 TO 764000. THE ASSIGNMENT OF ADDRESSES WILL START AT
THE HIGH ADDRESS LIMIT AND PROCEED DOWNWARD. USERS AND SPECIAL SYSTEMS
SHOULD START THEIR ASSIGNMENTS OF SPECIAL DEVICES AT THE LOW ADDRESS
LIMIT AND WORK UP. AFTER ASSIGNING ALL DR11A'S, ASSIGN DR11C'S

```
767776 TO 767770        DR11C #0     ;ASSUMING NO DR11A'S
767766 TO 767760        DR11C #1
        .
        .
767705 TO 767700        DR11C #7
        .
        .
767606 TO 767600        DR11C #15
```

8. MISCELLANEOUS

WHERE THERE ARE MULTIPLE DR11C OR A SYSTEM AND IT IS DESIRED TO TEST ONLY ONE OF THEM. THIS MAY BE ACHIEVED BY USING THE SPECIAL STARTING ADDRESS AND PLACING SW10 ON A ONE (UP) TO INHIBIT SEQUENCING TO THE NEXT DR11C. SEE 4.3.1.

8.1 EXECUTION TIME

FOR EACH DR11C ABOUT 1 MINUTE

8.2 UNTESTED LOGIC

SIGNALS TO USER NOT TESTED:
"NEW DATA READY"
"DATA TRANSMITTED"
"INIT" TO THE USER

9. PROGRAM DESCRIPTION

THIS PROGRAM WHEN STARTED AT 200 CHECKS THE STANDARD DR11-C'S

THE PROGRAM THEN PERFORMES AN INCREMENTAL LOGIC CHECK FOR THE SELECTED DR11C.

THE DATA REGISTER IS TESTED TO SEE IF "RESET" CLEARS IT, AND IF IT WILL HOLD ALL COMBINATIONS OF NUMBERS.

THE READ/WRITE BITS OF THE STATUS REGISTER ARE ALSO TESTED.

BOTH THE "A" AND "B" INTERRUPTS ARE TESTED TO SEE IF THEY INTERRUPT AT THE CORRECT BUS REQUEST LEVEL BR-5.

AT THE END OF THE TEST AN '/' IS TYPED AND ALSO THE ADDRESSES OF THE DR11-C CONTROL STATUS REGISTER AND IT'S SIDE INTERRUPT VECTOR IS TYPED (IF SELECTED VIA SWITCH 9.). THE PROGRAM THEN RETESTS THE UNIT (IF SELECTED VIA SWITCH 10) OR SCANS TO THE NEXT DR11-C. IF ANOTHER DR11-C IS ON THE SYSTEM THEN THE PROGRAM RESTARTS TESTING THE NEW DR11-C.

AFTER ALL DR11-C'S HAVE BEEN TESTED THE PROGRAM WILL TYPE '/' AND RESTART TESTING WITH THE INITIAL DR11-C.

IF NO ERRORS OCCUR AND THREE DR11-C'S ARE AVAILABLE AND SWITCH 9 IS DOWN THE PROGRAM WILL TYPE.

```
        160000 770 *
        157770 1000 *
        157760 1010 *
        /
        ETC.
```

IF SWITCH 9 IS UP THEN

```
        *
        *
        *
```

IF A POWER FAIL OCCURS THE PROGRAM WILL RESTART AT "START".

10. LISTING

```
        ;GENERAL REGISTER LOGIC TEST

        PSW=177776
        HLT=104000
        SR=177570
        CSR=167770
        STKPTR=1200
        ;REGISTER DEFINITIONS
                R0=R0
                R1=R1
                R2=R2
                R3=R3
                R4=R4
                R5=R5
                SP=R6
                PC=R7

       ;SWITCHES

                SW9=1000
                SW10=2000
                SW11=4000
                SW13=20000
                SW14=40000


                                .ENABLE ABS,AMA
                                .MCALL  .$TYPE,.$TYPOCT,.$TRAP,.EQUAT,.$POWER
                                .MCALL SETUP,.SETUP,.$CATCH

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
        001100                  STACK=  1100
                                .EQUIV  EMT,ERROR               ;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE               ;BASIC DEFINITION OF SCOPE CALL
        177776                  PS=     177776                  ;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
        177774                  STKLMT= 177774                  ;STACK LIMIT REGISTER
        177772                  PIRQ=   177772                  ;PROGRAM INTERRUPT REQUEST REGISTER
        177570                  SWR=    177570                  ;SWITCH REGISTER
        177570                  DISPLAY=SWR

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
        000000                  R0=     %0                      ;GENERAL REGISTER
        000001                  R1=     %1                      ;GENERAL REGISTER
        000002                  R2=     %2                      ;GENERAL REGISTER
        000003                  R3=     %3                      ;GENERAL REGISTER
        000004                  R4=     %4                      ;GENERAL REGISTER
        000005                  R5=     %5                      ;GENERAL REGISTER
        000006                  R6=     %6                      ;GENERAL REGISTER
        000007                  R7=     %7                      ;GENERAL REGISTER
                                .EQUIV  R6,SP                   ;STACK POINTER
                                .EQUIV  R7,PC                   ;PROGRAM COUNTER
```

```
                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                 100000         SW15=   100000
                 040000         SW14=   40000
                 020000         SW13=   20000
                 010000         SW12=   10000
                 004000         SW11=   4000
                 002000         SW10=   2000
                 001000         SW09=   1000
                 000400         SW08=   400
                 000200         SW07=   200
                 000100         SW06=   100
                 000040         SW05=   40
                 000020         SW04=   20
                 000010         SW03=   10
                 000004         SW02=   4
                 000002         SW01=   2
                 000001         SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                 100000         BIT15=  100000
                 040000         BIT14=  40000
                 020000         BIT13=  20000
                 010000         BIT12=  10000
                 004000         BIT11=  4000
                 002000         BIT10=  2000
                 001000         BIT09=  1000
                 000400         BIT08=  400
                 000200         BIT07=  200
                 000100         BIT06=  100
                 000040         BIT05=  40
                 000020         BIT04=  20
                 000010         BIT03=  10
                 000004         BIT02=  4
                 000002         BIT01=  2
                 000001         BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0
```

```
                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
            000004                      ERRVEC= 4               ;TIME OUT AND OTHER ERRORS
            000010                      RESVEC= 10              ;RESERVED AND ILLEGAL INSTRUCTIONS
            000014                      TBITVEC=14              ;"T" BIT
            000014                      TRTVEC= 14              ;TRACE TRAP
            000014                      BPTVEC= 14              ;BREAKPOINT TRAP (BPT)
            000020                      IOTVEC= 20              ;INPUT/OUTPUT TRAP (IOT) **SCOPE**
            000024                      PWRVEC= 24              ;POWER FAIL
            000030                      EMTVEC= 30              ;EMULATOR TRAP (EMT) **ERROR**
            000034                      TRAPVEC=34              ;"TRAP" TRAP
            000060                      TKVEC=  60              ;TTY KEYBOARD VECTOR
            000064                      TPVEC=  64              ;TTY PRINTER VECTOR
            000240                      PIRQVEC=240             ;PROGRAM INTERRUPT REQUEST VECTOR
            000000                      $TN=0
            000000                      $SWR=0

            000000                              .=0
  000000    000002                              .+2
  000002    000000                              HALT
  000004    000006                              .+2
  000006    000000                              HALT
  000010    000012                              .+2
  000012    000000                              HALT
  000014    000016                              .+2
  000016    000000                              HALT
  000020    000022                              .+2
  000022    000000                              HALT
  000024    000026                              .+2
  000026    000000                              HALT
  000030    000032                              .+2
  000032    000000                              HALT
  000034    000036                              .+2
  000036    000000                              HALT
  000040    000042                              .+2
  000042    000000                              HALT
  000044    000046                              .+2
  000046    000000                              HALT
  000050    000052                              .+2
  000052    000000                              HALT
  000054    000056                              .+2
  000056    000000                              HALT
  000060    000062                              .+2
  000062    000000                              HALT
  000064    000066                              .+2
  000066    000000                              HALT
  000070    000072                              .+2
  000072    000000                              HALT
  000074    000076                              .+2
  000076    000000                              HALT
  000100    000102                              .+2
  000102    000000                              HALT
  000104    000106                              .+2
  000106    000000                              HALT
  000110    000112                              .+2
  000112    000000                              HALT
  000114    000116                              .+2
```

```
  560  000116  000000                    HALT
  561  000120  000122                    .+2
  562  000122  000000                    HALT
  563  000124  000126                    .+2
  564  000126  000000                    HALT
  565  000130  000132                    .+2
  566  000132  000000                    HALT
  567  000134  000136                    .+2
  568  000136  000000                    HALT
  569  000140  000142                    .+2
  570  000142  000000                    HALT
  571  000144  000146                    .+2
  572  000146  000000                    HALT
  573  000150  000152                    .+2
  574  000152  000000                    HALT
  575  000154  000156                    .+2
  576  000156  000000                    HALT
  577  000160  000162                    .+2
  578  000162  000000                    HALT
  579  000164  000166                    .+2
  580  000166  000000                    HALT
  581  000170  000172                    .+2
  582  000172  000000                    HALT
  583  000174  000176                    .+2
  584  000176  000000                    HALT
  585  000200  000202                    .+2
  586  000202  000000                    HALT
  587  000204  000206                    .+2
  588  000206  000000                    HALT
  589  000210  000212                    .+2
  590  000212  000000                    HALT
  591  000214  000216                    .+2
  592  000216  000000                    HALT
  593  000220  000222                    .+2
  594  000222  000000                    HALT
  595  000224  000226                    .+2
  596  000226  000000                    HALT
  597  000230  000232                    .+2
  598  000232  000000                    HALT
  599  000234  000236                    .+2
  600  000236  000000                    HALT
  601  000240  000242                    .+2
  602  000242  000000                    HALT
  603  000244  000246                    .+2
  604  000246  000000                    HALT
  605  000250  000252                    .+2
  606  000252  000000                    HALT
  607  000254  000256                    .+2
  608  000256  000000                    HALT
  609  000260  000262                    .+2
  610  000262  000000                    HALT
  611  000264  000266                    .+2
  612  000266  000000                    HALT
  613  000270  000272                    .+2
  614  000272  000000                    HALT
  615  000274  000276                    .+2
```

```
  616  000276  000000                    HALT
  617  000300  000302                    .+2
  618  000302  000000                    HALT
  619  000304  000306                    .+2
  620  000306  000000                    HALT
  621  000310  000312                    .+2
  622  000312  000000                    HALT
  623  000314  000316                    .+2
  624  000316  000000                    HALT
  625  000320  000322                    .+2
  626  000322  000000                    HALT
  627  000324  000326                    .+2
  628  000326  000000                    HALT
  629  000330  000332                    .+2
  630  000332  000000                    HALT
  631  000334  000336                    .+2
  632  000336  000000                    HALT
  633  000340  000342                    .+2
  634  000342  000000                    HALT
  635  000344  000346                    .+2
  636  000346  000000                    HALT
  637  000350  000352                    .+2
  638  000352  000000                    HALT
  639  000354  000356                    .+2
  640  000356  000000                    HALT
  641  000360  000362                    .+2
  642  000362  000000                    HALT
  643  000364  000366                    .+2
  644  000366  000000                    HALT
  645  000370  000372                    .+2
  646  000372  000000                    HALT
  647  000374  000376                    .+2
  648  000376  000000                    HALT
  649  000400  000402                    .+2
  650  000402  000000                    HALT
  651  000404  000406                    .+2
  652  000406  000000                    HALT
  653  000410  000412                    .+2
  654  000412  000000                    HALT
  655  000414  000416                    .+2
  656  000416  000000                    HALT
  657  000420  000422                    .+2
  658  000422  000000                    HALT
  659  000424  000426                    .+2
  660  000426  000000                    HALT
  661  000430  000432                    .+2
  662  000432  000000                    HALT
  663  000434  000436                    .+2
  664  000436  000000                    HALT
  665  000440  000442                    .+2
  666  000442  000000                    HALT
  667  000444  000446                    .+2
  668  000446  000000                    HALT
  669  000450  000452                    .+2
  670  000452  000000                    HALT
  671  000454  000456                    .+2
```

```
   672  000456  000000                          HALT
   673  000460  000462                          .+2
   674  000462  000000                          HALT
   675  000464  000466                          .+2
   676  000466  000000                          HALT
   677  000470  000472                          .+2
   678  000472  000000                          HALT
   679  000474  000476                          .+2
   680  000476  000000                          HALT
   681  000500  000502                          .+2
   682  000502  000000                          HALT
   683  000504  000506                          .+2
   684  000506  000000                          HALT
   685  000510  000512                          .+2
   686  000512  000000                          HALT
   687  000514  000516                          .+2
   688  000516  000000                          HALT
   689  000520  000522                          .+2
   690  000522  000000                          HALT
   691  000524  000526                          .+2
   692  000526  000000                          HALT
   693  000530  000532                          .+2
   694  000532  000000                          HALT
   695  000534  000536                          .+2
   696  000536  000000                          HALT
   697  000540  000542                          .+2
   698  000542  000000                          HALT
   699  000544  000546                          .+2
   700  000546  000000                          HALT
   701  000550  000552                          .+2
   702  000552  000000                          HALT
   703  000554  000556                          .+2
   704  000556  000000                          HALT
   705  000560  000562                          .+2
   706  000562  000000                          HALT
   707  000564  000566                          .+2
   708  000566  000000                          HALT
   709  000570  000572                          .+2
   710  000572  000000                          HALT
   711  000574  000576                          .+2
   712  000576  000000                          HALT
   713  000600  000602                          .+2
   714  000602  000000                          HALT
   715  000604  000606                          .+2
   716  000606  000000                          HALT
   717  000610  000612                          .+2
   718  000612  000000                          HALT
   719  000614  000616                          .+2
   720  000616  000000                          HALT
   721  000620  000622                          .+2
   722  000622  000000                          HALT
   723  000624  000626                          .+2
   724  000626  000000                          HALT
   725  000630  000632                          .+2
   726  000632  000000                          HALT
   727  000634  000636                          .+2
```

```
  728  000636  000000                    HALT
  729  000640  000642                    .+2
  730  000642  000000                    HALT
  731  000644  000646                    .+2
  732  000646  000000                    HALT
  733  000650  000652                    .+2
  734  000652  000000                    HALT
  735  000654  000656                    .+2
  736  000656  000000                    HALT
  737  000660  000662                    .+2
  738  000662  000000                    HALT
  739  000664  000666                    .+2
  740  000666  000000                    HALT
  741  000670  000672                    .+2
  742  000672  000000                    HALT
  743  000674  000676                    .+2
  744  000676  000000                    HALT
  745  000700  000702                    .+2
  746  000702  000000                    HALT
  747  000704  000706                    .+2
  748  000706  000000                    HALT
  749  000710  000712                    .+2
  750  000712  000000                    HALT
  751  000714  000716                    .+2
  752  000716  000000                    HALT
  753  000720  000722                    .+2
  754  000722  000000                    HALT
  755  000724  000726                    .+2
  756  000726  000000                    HALT
  757  000730  000732                    .+2
  758  000732  000000                    HALT
  759  000734  000736                    .+2
  760  000736  000000                    HALT
  761  000740  000742                    .+2
  762  000742  000000                    HALT
  763  000744  000746                    .+2
  764  000746  000000                    HALT
  765  000750  000752                    .+2
  766  000752  000000                    HALT
  767  000754  000756                    .+2
  768  000756  000000                    HALT
  769  000760  000762                    .+2
  770  000762  000000                    HALT
  771  000764  000756                    .+2
  772  000766  000000                    HALT
  773  000770  000772                    .+2
  774  000772  000000                    HALT
  775  000774  000776                    .+2
  776  000776  000000                    HALT
  777          000020                    .=20
  778  000020  004264                    .SCOPE
  779  000022  000340                    340
  780  000024  004524                    PFAIL
  781  000026  000340                    340
  782  000030  003626                    .HLT
  783  000032  000340                    340
```

```
  784  000034  005462                          $TRAP
  785  000036  000340                          340
  786          000046                          .=46
  787  000046  003610                          LOCIC
  788          000200                          .=200
  789  000200  000137  001242                  JMP     @#START1        ;INITIAL START
  790  000204  000137  001256                  JMP     @#SPEC          ;TO SELECT UNIQUE ADDRESS AND VECTOR
  791  000210  000137  001366                  JMP     @#START         ;RESTART
  792          000250                          .=250
  793  000250  000000              EOPHLT:     HALT                    ; THIS IS AN END OF PASS HALT
  794                                                                  ; NOT AN ERROR HALT.
  795                                                                  ; PRESS CONTINUE TO CONTINUE THE
  796                                                                  ; PROGRAM.
  797
  798  000252  000207                          RTS     PC
  799          001200                          .=1200
  800
  801                              ;THIS TABLE CONTAINS INITIAL REGISTER AND VECTOR ADDRESSES
  802
  803  001200  167770              RCSR:       CSR
  804  001202  167772                          CSR+2
  805  001204  167774                          CSR+4
  806  001206  167773                          CSR+3
  807  001210  000300              RCSR1:      300
  808  001212  000302                          302
  809  001214  000304                          304
  810
  811                              ;THIS TABLE CONTAINS REGISTER AND VECTOR ADDRESSES OF THE DR11-C UNDER TEST
  812
  813  001216  167770              DRCSR:      167770                  ;ADDRESS OF DR11-C STATUS REGISTER
  814  001220  167772              DROBUF:     167772                  ;ADDRESS OF DR OUTPUT BUFFER REG.
  815  001222  167774              DRIBUF:     167774                  ;ADDRESS OF DR INPUT BUFFER REG.
  816  001224  167773              DRBHIO:     167773                  ;HIGH BYTE OF OUTPUT BUFFER REG.
  817
  818  001226  000300              DRVECA:     300                     ;INTERRUPT VECTOR OF UNIT UNDER TEST
  819  001230  000302              DRLVL:      302
  820  001232  000304              DRVECB:     304                     ;INTERRUPT VECTOR
  821  001234  000000              XORFLG:     0
  822
  823  001236  000000              COUNT:      0                       ;COUNT LOCATION
  824  001240  000240              PL:         240                     ;PRIORITY LEVEL
  825
  826  001242  012706  001200      START1:     MOV     #STKPTR,R6
  827  001246  004737  001272                  JSR     PC,FIRST
  828  001252  000137  001366                  JMP     @#START
  829  001256  012706  001200      SPEC:       MOV     #STKPTR,R6
  830  001262  004737  001272                  JSR     PC,FIRST
  831  001266  000137  004612                  JMP     SPECO
  832  001272  013746  000004      FIRST:      MOV     @#4,-(R6)
  833  001276  012737  001352  000004          MOV     #XORA,@#4
  834  001304  012737  000031  177060          MOV     #31,@#177060
  835  001312  012637  000004                  MOV     (R6)+,@#4
  836  001316  012737  177777  001234          MOV     #-1,@#XORFLG
  837  001324  012701  160000                  MOV     #160000,R1
  838  001330  004737  004642                  JSR     PC,@#SPEC1
  839  001334  012701  000770                  MOV     #770,R1
```

```
 940  001340  004737  004672          JSR     PC,@#SPEC2
 941  001344  104400                  TYPE
 942  001346  004712                  MESS1
 943  001350  000207                  RTS     PC
 944  001352  022626          XORA:   CMP     (R6)+,(R6)+
 945  001354  012637  000004          MOV     (R6)+,@#4
 946  001360  005037  001234          CLR     @#XORFLG
 947  001364  000207                  RTS     PC
 948                          ;INITIALIZE ADDRESS AND VECTORS
 949  001366  012700  001200  START:  MOV     #RCSR,R0        ;GET ADDRESS OF FIRST POSSIBLE DR11-C'S
 950  001372  012701  001216          MOV     #DRCSR,R1
 951  001376  012021                  MOV     (R0)+,(R1)+     ;LOAD INITIAL TEST ADDRESSES
 952  001400  012021                  MOV     (R0)+,(R1)+
 953  001402  012021                  MOV     (R0)+,(R1)+
 954  001404  012021                  MOV     (R0)+,(R1)+
 955  001406  012021                  MOV     (R0)+,(R1)+
 956  001410  012021                  MOV     (R0)+,(R1)+
 957  001412  012021                  MOV     (R0)+,(R1)+
 958  001414  012706  001200  RSTART: MOV     #STKPTR,R6      ;SET UP STACK
 959  001420  012737  001446  004372  MOV     #BEGIN,RETURN   ;SET SCOPE RETURN
 960  001426  005037  004370          CLR     @#SCOPEF
 961
 962                          ;DOES RESET CLEAR REGISTER?
 963  001432  032737  001000  177570  BIT     #SW9,@#SR
 964  001440  001002                  BNE     BEGIN
 965  001442  004737  004410          JSR     PC,@#MOREID
 966  001446  013705  001216  BEGIN:  MOV     DRCSR,R5        ;GET ADDRESS OF STATUS REGISTER
 967  001452  012777  000240  176316  MOV     #240,@#PSW      ;SET PRIORITY LEVEL 6
 968  001460  012737  001516  000004  MOV     #1$,@#4         ;SET TIME OUT TRAP VECTOR
 969  001466  012737  000010  004366  MOV     #10,ICOUNT
 970  001474  012777  177777  177516  MOV     #-1,@DROBUF     ;PRESET OUTPUT BUFFER
 971  001502  000005                  RESET                   ;CLEAR DATA REGISTER
 972  001504  017700  177510          MOV     @DROBUF,R0      ;GET RESULT OF RESET
 973  001510  001403                  BEQ     2$
 974  001512  104000                  HLT                     ;DATA REGISTER NOT CLEAR
 975  001514  000401                  BR      2$
 976  001516  104000          1$:     HLT                     ;ERROR! TIMED OUT WHEN REFERENCING DROBUF.
 977  001520  012706  001200  2$:     MOV     #STKPTR,SP      ;RESET STACK POINTER
 978  001524  012737  000006  000004  MOV     #6,@#4          ;RESTORE TIME OUT TRAP
 979
 980  001532  000004                  SCOPE
 981  001534  012737  004000  004366  MOV     #4000,ICOUNT
 982  001542  012777  177777  177450  MOV     #-1,@DROBUF     ;ALL ONES TO REGISTER
 983  001550  017700  177444          MOV     @DROBUF,R0
 984  001554  022700  177777          CMP     #-1,R0
 985  001560  001401                  BEQ     .+4
 986  001562  104000                  HLT                     ;REG WILL NOT HOLD ONES
 987
 988  001564  000004                  SCOPE
 989  001566  012737  000010  004366  MOV     #10,ICOUNT
 990  001574  012777  177777  177416  MOV     #-1,@DROBUF
 991  001602  000005                  RESET                   ;SET DATA TO ALL ONES
 992  001604  005777  177412          TST     @DRIBUF         ;SHOULD CLEAR REGISTER
 993  001610  001401                  BEQ     .+4
 994  001612  104000                  HLT                     ;REG FAILED TO CLEAR
 995
```

```
 896  001614  000004                          SCOPE
 897  001616  012737  004000  004366          MOV     #4000,ICOUNT
 898  001624  012777  052525  177366          MOV     #52525,@DROBUF
 899  001632  017700  177362                  MOV     @DROBUF,R0
 900  001636  022700  052525                  CMP     #52525,R0
 901  001642  001401                          BEQ     .+4
 902  001644  104000                          HLT                     ;DATA NOT=52525
 903
 904  001646  000004                          SCOPE
 905  001650  012777  125252  177342          MOV     #125252,@DROBUF
 906  001656  017700  177336                  MOV     @DROBUF,R0
 907  001662  022700  125252                  CMP     #125252,R0
 908  001666  001401                          BEQ     .+4
 909  001670  104000                          HLT                     ;DATA NOT=125252
 910
 911                                  ;TEST RELIABILITY OF DR11-C OUTPUT BUFFER REGISTER
 912  001672  000004                          SCOPE
 913  001674  012737  000040  004366          MOV     #40,@#ICOUNT
 914  001702  010502                  BUFTST: MOV     R5,R2           ;GET ADDRESS OF DRCSR
 915  001704  005722                          TST     (R2)+           ;R2=ADDRESS OF OUTPUT BUFFER REG.
 916  001706  012703  000401                  MOV     #401,R3         ;LOAD CONSTANT
 917  001712  012704  000400          1$:     MOV     #256.,R4        ;SET COUNT
 918  001716  005000                          CLR     R0              ;PRESET EXPECTED RESULT
 919  001720  005012                          CLR     (R2)            ;CLEAR REGISTER
 920  001722  060300                  2$:     ADD     R3,R0
 921  001724  060312                          ADD     R3,(R2)
 922  001726  021200                          CMP     (R2),R0
 923  001730  001401                          BEQ     .+4
 924  001732  104000                          HLT
 925  001734  005304                          DEC     R4
 926  001736  001371                          BNE     2$
 927  001740  006303                          ASL     R3
 928  001742  001363                          BNE     1$
 929
 930                                  ;TEST THAT BYTE REFERENCE TO DROBUF AFFECT PROPER BYTE ONLY
 931
 932  001744  000004                  TAG:    SCOPE
 933  001746  012777  177777  177244          MOV     #-1,@DROBUF
 934  001754  105077  177240                  CLRB    @DROBUF         ;CLEAR LOW BYTE
 935  001760  017700  177234                  MOV     @DROBUF,R0
 936  001764  022700  177400                  CMP     #177400,R0
 937  001770  001401                          BEQ     .+4
 938  001772  104000                          HLT                     ;BYTE LOW FAILED TO CLEAR
 939
 940  001774  000004                          SCOPE
 941  001776  012777  177777  177214          MOV     #-1,@DROBUF
 942  002004  105077  177214                  CLRB    @DRBHIO         ;CLEAR HIGH BYTE
 943  002010  017700  177204                  MOV     @DROBUF,R0
 944  002014  022700  000377                  CMP     #377,R0
 945  002020  001401                          BEQ     .+4
 946  002022  104000                          HLT                     ;HIGH BYTE CLEAR FAILED
 947
 948  002024  000004                          SCOPE
 949  002026  005037  002076                  CLR     @#2$
 950  002032  012704  002076                  MOV     #2$,R4
 951  002036  005077  177156                  CLR     @DROBUF
```

```
 952  002042  105077  177156              CLRB    @DRBHIO
 953  002046  105277  177152      1$:     INCB    @DRBHIO          ;INCREMENT HIGH BYTE
 954  002052  105264  000001              INCB    1(R4)
 955  002056  027714  177136              CMP     @DROBUF,(R4)
 956  002062  001401                      BEQ     .+4
 957  002064  104000                      HLT                      ;HIGH BYTE HAS BAD DATA
 958  002066  105764  000001              TSTB    1(R4)
 959  002072  001402                      BEQ     3$
 960  002074  000764                      BR      1$
 961  002076  000000              2$:     .WORD   0
 962  002100  000004              3$:     SCOPE
 963                              ;CONTROL STATUS REGISTER (DRCSR) TESTS.
 964  002102  005015                      CLR     (R5)
 965  002104  011500                      MOV     (R5),R0
 966  002106  001401                      BEQ     .+4
 967  002110  104000                      HLT
 968  002112  012715  000140              MOV     #140,@R5         ;INTERRUPT ENABLE FOR A+B
 969  002116  011500                      MOV     @R5,R0
 970  002120  022700  000140              CMP     #140,R0          ;ENABLE BITS
 971  002124  001401                      BEQ     .+4
 972  002126  104000                      HLT
 973
 974  002130  000004                      SCOPE
 975  002132  012737  000010  004366      MOV     #10,ICOUNT
 976  002140  012715  000140              MOV     #140,@R5         ;SET INTERRUPT ENABLE FLOPS
 977  002144  000005                      RESET                    ;CLEAR THOSE FLOPS
 978  002146  011500                      MOV     @R5,R0
 979  002150  001401                      BEQ     .+4
 980  002152  104000                      HLT                      ;RESET DID NOT CLEAR INTERRUPT ENABLE BITS
 981
 982  002154  000004                      SCOPE
 983  002156  052715  000001              BIS     #1,@R5           ;SHOULD SET REQ A ALSO
 984  002162  021527  000201              CMP     @R5,#201
 985  002166  001401                      BEQ     .+4
 986  002170  104000                      HLT
 987  002172  005015                      CLR     @R5
 988
 989  002174  000004                      SCOPE
 990  002176  052715  000002              BIS     #2,@R5           ;SHOULD SET REQ B
 991  002202  021527  100002              CMP     @R5,#100002
 992  002206  001401                      BEQ     .+4
 993  002210  104000                      HLT
 994  002212  005015                      CLR     @R5
 995
 996  002214  000004                      SCOPE
 997  002216  052737  000340  177776      BIS     #340,@#PSW
 998  002224  052715  177777              BIS     #-1,@R5
 999  002230  022715  100343              CMP     #100343,(R5)
1000  002234  001401                      BEQ     .+4
1001  002236  104000                      HLT
1002  002240  042715  000003              BIC     #3,@R5
1003  002244  022715  000140              CMP     #140,@R5
1004  002250  001401                      BEQ     .+4
1005  002252  104000                      HLT                      ;WRONG BITS SET
1006
1007  002254  000004                      SCOPE
```

```
 1008  002256  012737  000340  177776          MOV     #340,@#PSW
 1009  002264  052715  000003                  BIS     #3,@R5
 1010  002270  000005                          RESET
 1011  002272  005715                          TST     @R5
 1012  002274  001401                          BEQ     .+4
 1013  002276  104000                          HLT                     ;RESET DID NOT CLEAR
 1014
 1015  002300  000004                          SCOPE
 1016  002302  012737  004000  004366          MOV     #4000,ICOUNT
 1017  002310  005015                          CLR     @R5
 1018  002312  005215                          INC     @R5
 1019  002314  105715                          TSTB    @R5
 1020  002316  100401                          BMI     .+4
 1021  002320  104000                          HLT                     ;BIT 0 DID NOT SET BIT 7
 1022
 1023  002322  000004                          SCOPE
 1024  002324  012715  000002                  MOV     #2,@R5
 1025  002330  005715                          TST     @R5
 1026  002332  100401                          BMI     .+4
 1027  002334  104000                          HLT                     ;BIT 1 DID NOT SET BIT 15
 1028
 1029                                  ;TEST EXTERNAL TRANSFERS - CABLE MUST BE CONNECTED
 1030  002336  000004                          SCOPE
 1031  002340  005077  176654                  CLR     @DROBUF
 1032  002344  017777  176652  176646          MOV     @DRIBUF,@DROBUF ;TEST TRANSFER OF ZERO
 1033  002352  001401                          BEQ     .+4
 1034  002354  104000                          HLT                     ;NOT EQUAL TO ZERO
 1035
 1036  002356  000004                          SCOPE
 1037  002360  012777  177777  176632          MOV     #-1,@DROBUF
 1038  002366  017777  176630  176624          MOV     @DRIBUF,@DROBUF ;MOV ALL ONES
 1039  002374  022777  177777  176616          CMP     #-1,@DROBUF
 1040  002402  001401                          BEQ     .+4
 1041  002404  104000                          HLT                     ;NOT ALL ONES
 1042
 1043  002406  000004                          SCOPE
 1044  002410  005037  004366                  CLR     ICOUNT
 1045  002414  005000                          CLR     R0
 1046  002416  010077  176576          TST6:   MOV     R0,@DROBUF      ;TEST ALL NUMBERS
 1047  002422  017777  176574  176570          MOV     @DRIBUF,@DROBUF
 1048  002430  020077  176564                  CMP     R0,@DROBUF
 1049  002434  001401                          BEQ     .+4
 1050  002436  104000                          HLT                     ;ERROR - CHECK R0 FOR GOOD
 1051  002440  005200                          INC     R0              ;DROBUF FOR BAD
 1052  002442  001403                          BEQ     TST9
 1053  002444  005077  176550                  CLR     @DROBUF
 1054  002450  000762                          BR      TST6
 1055  002452  000004          TST9:           SCOPE
 1056  002454  012737  000005  004366          MOV     #5,@#ICOUNT
 1057                                  ;TEST DATA FROM BLACK BOX (NOT CONNECTED)
 1058  002462  012777  177777  176530          MOV     #-1,@DROBUF
 1059  002470  017777  176524  176524          MOV     @DROBUF,@DRIBUF ;STATIC LINES EQUAL ONES
 1060  002476  017700  176520                  MOV     @DRIBUF,R0      ;DATA REGISTER TO R0
 1061  002502  022700  177777                  CMP     #-1,R0
 1062  002506  001401                          BEQ     .+4
 1063  002510  104000                          HLT                     ;REG 0 SHOULD = ALL ONES
```

```
1064
1065                                        ;READY BIT IS IN A ONE STATE
1066  002512  000004                               SCOPE
1067  002514  012715  000003                       MOV     #3,@R5          ;CSR0 AND CSR1
1068  002520  011500                               MOV     (R5),R0
1069  002522  022700  100203                       CMP     #100203,R0
1070  002526  001401                               BEQ     .+4
1071  002530  104000                               HLT
1072
1073                                        ;CAN WE RAISE INTERUPT "A"
1074  002532  000004                               SCOPE
1075  002534  052737  000340  177776               BIS     #340,@#PSW      ;LOCK OUT INTERRUPTS
1076  002542  012706  001200                       MOV     #STKPTR,R6
1077  002546  012777  002570  176452               MOV     #TST4,@DRVECA   ;INTERRUPT RETURN POINTER
1078  002554  012715  000101                       MOV     #101,@R5        ;INTERRUPT ENABLE AND CSR0
1079  002560  005037  177776                       CLR     @#PSW
1080  002564  000240                               NOP
1081  002566  104000                               HLT                     ;NO "A" INTERRUPT
1082  002570  005015                       TST4:   CLR     @R5
1083  002572  013777  001230  176426               MOV     DRLVL,@DRVECA   ;MOVE .+2 TO "A" INTERRUPT VECTOR
1084
1085                                        ;RAISE INTERRUPT "B"
1086  002600  000004                               SCOPE
1087  002602  012706  001200                       MOV     #STKPTR,R6
1088  002606  052737  000340  177776               BIS     #340,@#PSW
1089  002614  012777  002640  176410               MOV     #TST5,@DRVECB
1090  002622  012715  000042                       MOV     #42,@R5         ;IE AND CSR1
1091  002626  042737  000377  177776               BIC     #377,@#PSW
1092  002634  000240                               NOP
1093  002636  104000                               HLT                     ;NO B INTERRUPT
1094  002640  005015                       TST5:   CLR     @R5
1095
1096                                        ;TEST FOR INTERRUPT FROM DEVICE
1097  002642  013777  001240  176360               MOV     PL,@DRLVL
1098  002650  042737  000340  177776               BIC     #340,@#PSW      ;PROCESSOR LEVEL ZERO
1099  002656  012777  002710  176342               MOV     #TINT1,@DRVECA
1100  002664  012706  001200                       MOV     #STKPTR,R6      ;STACK POINTER
1101  002670  042777  000100  176320               BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
1102  002676  052777  000101  176312               BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
1103  002704  000240                               NOP
1104  002706  104000                               HLT                     ;NO DEVICE INTERRUPT OCCURED
1105  002710  000004                       TINT1:  SCOPE
1106
1107                                        ;TEST FOR INTERRUPT FROM THE DEVICE
1108  002712  042737  000340  177776               BIC     #340,@#PSW
1109  002720  052737  000040  177776               BIS     #040,@#PSW      ;SET TO PRIORITY LEVEL 1
1110  002726  012777  002760  176272               MOV     #TINT2,@DRVECA  ;INTERRUPT VECTOR ADDRESS
1111  002734  012706  001200                       MOV     #STKPTR,R6      ;SET UP STACK POINTER
1112  002740  042777  000100  176250               BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
1113  002746  052777  000101  176242               BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
1114  002754  000240                               NOP
1115  002756  104000                               HLT                     ;NO DEVICE INTERRUPT OCCURED
1116
1117  002760  000004                       TINT2:  SCOPE
1118  002762  042737  000340  177776               BIC     #340,@#PSW
1119  002770  052737  000100  177776               BIS     #100,@#PSW      ;SET TO PRIORITY LEVEL 2
```

```
1120  002776  012777  003030  176222          MOV     #TINT3,@DRVECA  ;INTERRUPT VECTOR ADDRESS
1121  003004  012706  001200                  MOV     #STKPTR,R6      ;SET UP STACK POINTER
1122  003010  042777  000100  176200          BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
1123  003016  052777  000101  176172          BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
1124  003024  000240                          NOP
1125  003026  104000                          HLT                     ;NO DEVICE INTERRUPT OCCURED
1126
1127  003030  000004                  TINT3:  SCOPE
1128                                  ;TEST FOR INTERRUPT FROM THE DEVICE
1129  003032  042737  000340  177776          BIC     #340,@#PSW
1130  003040  052737  000140  177775          BIS     #140,@#PSW      ;SET TO PRIORITY LEVEL 3
1131  003046  012777  003100  176152          MOV     #TINT4,@DRVECA  ;INTERRUPT VECTOR ADDRESS
1132  003054  012706  001200                  MOV     #STKPTR,R6      ;SET UP STACK POINTER
1133  003060  042777  000100  176130          BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
1134  003066  052777  000101  176122          BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
1135  003074  000240                          NOP
1136  003076  104000                          HLT                     ;NO DEVICE INTERRUPT OCCURED
1137  003100  000004                  TINT4:  SCOPE
1138
1139                                  ;TEST FOR INTERRUPT FROM DEVICE
1140  003102  042737  000340  177776          BIC     #340,@#PSW
1141  003110  052737  000200  177776          BIS     #200,@#PSW      ;RAISE PROCESSOR PRIORITY TO LEVEL 4
1142  003116  012777  003160  176102          MOV     #TINT5,@DRVECA  ;IN CASE OF INTERRUPT
1143  003124  012706  001200                  MOV     #STKPTR,R6      ;SET STACK POINTER
1144  003130  042777  000100  176060          BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
1145  003136  052777  000101  176052          BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE AND CSR0
1146  003144  000240                          NOP                     ;LET INTERRUPT OCCUR
1147  003146  042777  000100  176042          BIC     #100,@DRCSR
1148  003154  000240                          NOP
1149  003156  104000                          HLT                     ;NO DEVICE INTERRUPT OCCURED
1150  003160  000004                  TINT5:  SCOPE
1151
1152                                  ;TEST FOR NO INTERRUPT FROM DEVICE (HIGHEST PROCESSOR PRIORITY)
1153  003162  052737  000340  177776          BIS     #340,@#PSW      ;RAISE PROCESSOR PRIORITY TO HIGHEST LEVEL
1154  003170  012777  003230  176030          MOV     #TINT6,@DRVECA  ;IN CASE OF INTERRUPT
1155  003176  012706  001200                  MOV     #STKPTR,R6      ;SET STACK POINTER
1156  003202  042777  000100  176006          BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
1157  003210  052777  000101  176000          BIS     #101,@DRCSR
1158  003216  000240                          NOP
1159  003220  042777  000100  175770          BIC     #100,@DRCSR
1160  003226  000401                          BR      .+4             ;WITH NO INTERRUPT, BRANCH OVER HALT
1161  003230  104000                  TINT6:  HLT                     ;INTERRUPT OCCURED
1162  003232  000004                          SCOPE
1163
1164                                  ;TEST FOR NO INTERRUPT FROM DEVICE
1165  003234  042737  000340  177776          BIC     #340,@#PSW
1166  003242  052737  000240  177776          BIS     #240,@#PSW      ;RAISE PROCESSOR PRIORITY TO LEVEL 5
1167  003250  012777  003310  175750          MOV     #TINT7,@DRVECA  ;IN CASE OF INTERRUPT
1168  003256  012706  001200                  MOV     #STKPTR,R6      ;SET STACK POINTER
1169  003262  042777  000100  175726          BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
1170  003270  052777  000101  175720          BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE AND CSR0
1171  003276  000240                          NOP
1172  003300  042777  000100  175710          BIC     #100,@DRCSR     ;DON'T LEAVE IT SET
1173  003306  000401                          BR      .+4             ;WITH NO INTERRUPT, BRANCH OVER HALT
1174  003310  104000                  TINT7:  HLT                     ;INTERRUPT OCCURED
1175  003312  000004                          SCOPE
```

```
 1176
 1177                                        ;TEST FOR NO INTERRUPT FROM DEVICE
 1178  003314  042737  000340  177776              BIC     #340,@#PSW
 1179  003322  052737  000300  177776              BIS     #300,@#PSW      ;RAISE PROCESSOR PRIORITY TO LEVEL 6
 1180  003330  012777  003370  175670              MOV     #TINT8,@DRVECA  ;IN CASE OF INTERRUPT
 1181  003336  012706  001200                      MOV     #STKPTR,R6      ;SET STACK POINTER
 1182  003342  042777  000100  175646              BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
 1183  003350  052777  000101  175640              BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
 1184  003356  042777  000100  175632              BIC     #100,@DRCSR     ;DON'T LEAVE IT SET
 1185  003364  000240                              NOP
 1186  003366  000401                              BR      .+4             ;WITH NO INTERRUPT, BRANCH OVER HALT
 1187  003370  104000                      TINT8:  HLT                     ;INTERRUPT OCCURED
 1188  003372  000004                              SCOPE
 1189
 1190  003374  013777  001230  175624              MOV     DRLVL,@DRVECA   ;FOR FALSE INTERRUPT
 1191  003402  005077  175620                      CLR     @DRVECA
 1192
 1193                                        ;END OF TEST ROUTINE
 1194  003406  010237  003750              END:    MOV     R2,SAVR2
 1195  003412  010337  003752                      MOV     R3,SAVR3
 1196  003416  012702  000052                      MOV     #'*,R2
 1197  003422  012703  000062                      MOV     #50.,R3
 1198  003426  000005                      2$:     RESET
 1199  003430  005303                              DEC     R3
 1200  003432  001375                              BNE     2$
 1201  003434  032737  000100  177570              BIT     #SW06,SWR
 1202  003442  001402                              BEQ     1$
 1203  003444  004737  000250                      JSR     PC,EOPHLT
 1204  003450  104400  000002              1$:     TYPE    ,R2
 1205  003454  013702  003750                      MOV     SAVR2,R2
 1206  003460  013703  003752                      MOV     SAVR3,R3
 1207  003464  032737  002000  177570              BIT     #SW10,@#SR      ;LOOP ON SELECTED DR?
 1208  003472  001402                              BEQ     4$
 1209  003474  000137  001414                      JMP     @#RSTART        ;REPEAT TEST ON DR11C SELECTED
 1210                                        ;STEP TO NEXT DR11-C
 1211  003500  012700  000010              4$:     MOV     #10,R0          ;STEPPING CONSTANT
 1212  003504  012737  003554  000004              MOV     #5$,@#4         ;SET TIME OUT TRAP
 1213  003512  160005                              SUB     R0,R5           ;STEP TO NEXT DR11-C ADDRESS
 1214  003514  005715                              TST     (R5)            ;WILL TIME OUT IF NOT AVAILABLE
 1215  003516  012705  001216                      MOV     #DRCSR,R5       ;SET TABLE POINTER
 1216  003522  160025                              SUB     R0,(R5)+
 1217  003524  160025                              SUB     R0,(R5)+
 1218  003526  160025                              SUB     R0,(R5)+
 1219  003530  160025                              SUB     R0,(R5)+
 1220  003532  060025                              ADD     R0,(R5)+
 1221  003534  060025                              ADD     R0,(R5)+
 1222  003536  060025                              ADD     R0,(R5)+
 1223  003540  000137  001414                      JMP     @#RSTART        ;RESTART TEST USING NEXT DR11-C
 1224  003544  032737  001000  177570              BIT     #SW9,@#SR
 1225  003552  001013                              BNE     8$
 1226  003554  012777  000057  001304  5$:     MOV     #'/,@$TPB       ;TYPE '/'
 1227  003562  105777  001276              6$:     TSTB    @$TPS
 1228  003566  100375                              BPL     6$
 1229  003570  005077  001272                      CLR     @$TPB
 1230  003574  105777  001264              7$:     TSTB    @$TPS
 1231  003600  100375                              BPL     7$
```

```
 1232  003602  013700  000042          9$:     MOV     @#42,R0
 1233  003606  001405                          BEQ     END1
 1234  003610  004710                  LOGIC:  JSR     PC,(R0)
 1235  003612  000240                          NOP
 1236  003614  000240                          NOP
 1237  003616  000240                          NOP
 1238  003620  000240                          NOP
 1239  003622  000137  001366          END1:   JMP     @#START
 1240
 1241                                  ;ENTERED WITH SYSTEM TRAP CALL(HLT)
 1242                                  ;PRINT OUT THE ERROR PC AND STATUS REGISTER
 1243  003626  033727  177570  020000  .HLT:   BIT     SR,#SW13        ;TEST FOR INHIBIT PRINT OUT
 1244  003634  001401                          BEQ     .+4             ;BRANCH TO PRINT
 1245  003636  000002                          RTI                     ;INHIBIT, RETURN TO MAIN STREAM
 1246  003640  012637  017400                  MOV     (6)+,SAVPC      ;PC OF FAILING ROUTINE
 1247  003644  012637  017402                  MOV     (6)+,SAVCC      ;CC OF ERROR CONDITION
 1248  003650  024646                          CMP     -(6),-(6)       ;REPOSITION THE STACK
 1249  003652  104400  004747                  TYPE    ,MCRLF
 1250  003656  013746  017400                  MOV     SAVPC,-(SP)             ;SAVE SAVPC FOR TYPEOUT
 1251  003662  104404                          TYPOS                           ;GO TYPE--OCTAL ASCII
 1252  003664     006                          .BYTE   6                       ;TYPE 6 DIGITS
 1253  003665     000                          .BYTE   0                       ;SUPPRESS LEADING ZEROS
 1254  003666  104400  004752                  TYPE    ,MSPACE
 1255  003672  013746  017402                  MOV     SAVCC,-(SP)             ;SAVE SAVCC FOR TYPEOUT
 1256  003676  104404                          TYPOS                           ;GO TYPE--OCTAL ASCII
 1257  003700     006                          .BYTE   6                       ;TYPE 6 DIGITS
 1258  003701     000                          .BYTE   0                       ;SUPPRESS LEADING ZEROS
 1259  003702  104400  004752                  TYPE    ,MSPACE
 1260  003706  013746  001216                  MOV     DRCSR,-(SP)             ;SAVE DRCSR FOR TYPEOUT
 1261  003712  104404                          TYPOS                           ;GO TYPE--OCTAL ASCII
 1262  003714     006                          .BYTE   6                       ;TYPE 6 DIGITS
 1263  003715     000                          .BYTE   0                       ;SUPPRESS LEADING ZEROS
 1264  003716  104400  004752                  TYPE    ,MSPACE
 1265  003722  013746  001226                  MOV     DRVECA,-(SP)            ;SAVE DRVECA FOR TYPEOUT
 1266  003726  104404                          TYPOS                           ;GO TYPE--OCTAL ASCII
 1267  003730     006                          .BYTE   6                       ;TYPE 6 DIGITS
 1268  003731     000                          .BYTE   0                       ;SUPPRESS LEADING ZEROS
 1269  003732  104400  004747                  TYPE    ,MCRLF
 1270  003736  005737  177570                  TST     SR
 1271  003742  100001                          BPL     .+4
 1272  003744  000000                          HALT                    ;HALT ON ERROR SET
 1273  003746  000002                          RTI                     ;RETURN TO MAIN STREAM
 1274  003750  000000                  SAVR2:  0
 1275  003752  000000                  SAVR3:  0
 1276  003754  000000                  SAVR4:  0
 1277
 1278  003756  005037  004234          PRTAB:  CLR     BINCT
 1279  003762  005037  004232                  CLR     WGTCT
 1280  003766  012704  004240                  MOV     #LIST,R4        ;GET LIST ADDRESS
 1281  003772  012737  000005  004236          MOV     #5,ASCNT
 1282  004000  012737  000007  004226          MOV     #7,SEVEN
 1283  004006  012737  000001  004230          MOV     #1,DECML
 1284  004014  105777  001044          WAIT1:  TSTB    @$TPS
 1285  004020  100375                          BPL     WAIT1
 1286  004022  005702                          TST     R2
 1287  004024  100404                          BMI     MINUS           ;NEG SIGN PRINT 1
```

```
 1288  004026  012777  000260  001032          MOV     #260,@$TPB      ;POS SIGN PRINT 0
 1289  004034  000403                          BR      STAR
 1290  004036  012777  000261  001022  MINUS:  MOV     #261,@$TPB
 1291  004044  013703  004226          STAR:   MOV     SEVEN,R3        ;PUT MASK IN R3
 1292  004050  010237  004224                  MOV     R2,TOODLE       ;GET READY TO DOODLE NUMBER IN TOODLE
 1293  004054  005137  004224                  COM     TOODLE          ;COMPENSATES FOR COMPLEMENT DURING BIC
 1294  004060  043703  004224                  BIC     TOODLE,R3       ;AND IN OCTAL CHARACTER
 1295  004064  001410                          BEQ     WRTOC           ;ZERO, WRITE 0 IN LIST
 1296  004066  063737  004230  004232  MKNUM:  ADD     DECML,WGTCT     ;COUNT UP TO
 1297  004074  005237  004234                  INC     BINCT           ;AND RECORD
 1298  004100  023703  004232                  CMP     WGTCT,R3        ;SAME BINARY WEIGHT
 1299  004104  001370                          BNE     MKNUM           ;KEEP COUNTN
 1300  004106  062737  000260  004234  WRTOC:  ADD     #260,BINCT      ;ADD ASCII PREFIX
 1301  004114  013724  004234                  MOV     BINCT,(4)+      ;WRITE ASCII CHAR IN LIST
 1302  004120  063737  004226  004230          ADD     SEVEN,DECML     ;EXPAND BINARY WEIGHT
 1303  004126  005037  004232                  CLR     WGTCT
 1304  004132  005037  004234                  CLR     BINCT
 1305  004136  005337  004236                  DEC     ASCNT
 1306  004142  001410                          BEQ     XLIST           ;5 CHAR IN LIST
 1307  004144  012703  000003                  MOV     #3,R3           ;SET X3 FOR ADD LOOP
 1308  004150  063737  004226  004226  MOADD:  ADD     SEVEN,SEVEN     ;MAKING SEVENTY BY SEVEN
 1309  004156  005303                          DEC     R3
 1310  004160  001373                          BNE     MOADD
 1311  004162  000730                          BR      STAR            ;NX SEVEN SET GET NX OCTAL
 1312  004164  012737  000005  004236  XLIST:  MOV     #5,ASCNT        ;SEND 5 CHAR TO TTY
 1313  004172  105777  000666          WAIT2:  TSTB    @$TPS
 1314  004176  100375                          BPL     WAIT2
 1315  004200  014477  000662                  MOV     -(4),@$TPB
 1316  004204  005337  004236                  DEC     ASCNT
 1317  004210  001401                          BEQ     HOFHM           ;FINISH PRINTING GET NXT NUM
 1318  004212  000767                          BR      WAIT2
 1319  004214  105777  000644          HOFHM:  TSTB @$TPS
 1320  004220  100375                          BPL     .-4
 1321  004222  000207                          RTS     R7              ;HEAD FOR HOME
 1322  004224  000000                  TOODLE: 0
 1323  004226  000000                  SEVEN:  0
 1324  004230  000000                  DECML:  0
 1325  004232  000000                  WGTCT:  0
 1326  004234  000000                  BINCT:  0
 1327  004236  000000                  ASCNT:  0
 1328  004240  000000                  LIST:   0
 1329  004242  000000                          0
 1330  004244  000000                          0
 1331  004246  000000                          0
 1332  004250  000000                          0
 1333                                  ;SCOPE LOOP ROUTINE ENTERED BY USER TRAP
 1334  004252  022606                  SCOPEB: CMP     (6)+,R6         ;REPOSITION THE STACK
 1335  004254  012637  177776                  MOV     (6)+,@#PSW
 1336  004260  000177  000106                  JMP     @RETURN         ;SCOPE RETURN
 1337
 1338                                  ;SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES
 1339  004264  032737  040000  177570  .SCOPE: BIT     #SW14,SR        ;TEST SR FOR SCOPE
 1340  004272  001367                          BNE     SCOPEB          ;YES SCOPE
 1341  004274  005737  001234                  TST     @#XORFLG
 1342  004300  100012                          BPL     1$
 1343  004302  013746  000004                  MOV     @#4,-(R6)
```

```
1344  004306  012737  004400  000004          MOV     #XOR,@#4
1345  004314  012737  000031  177060          MOV     #31,@#177060
1346  004322  012637  000004                  MOV     (R6)+,@#4
1347  004326  032737  004000  177570  1$:     BIT     #SW11,SR        ;NO - TEST FOR ITERATION
1348  004334  001011                          BNE     SCOPEA          ;INHIBIT ITERATION
1349  004336  023737  004370  004366          CMP     SCOPEF,ICOUNT
1350  004344  001403                          BEQ     SCOPEG          ;EXIT - DONE
1351  004346  005237  004370                  INC     SCOPEF          ;INCREMENT COUNT
1352  004352  000737                          BR      SCOPEB          ;LOOP SOME MORE
1353  004354  005037  004370          SCOPEG: CLR     SCOPEF          ;CLEAR COUNT
1354  004360  011637  004372          SCOPEA: MOV     @R6,RETURN      ;SAVE SCOPE RETURN POINTER
1355  004364  000002                          RTI                     ;RETURN INLINE-NEXT TEST
1356  004366  004000                  ICOUNT: 4000
1357  004370  000000                  SCOPEF: 0                       ;COUNT LOCATION FOR ITERATION LOOP
1358  004372  001446                  RETURN: BEGIN                   ;ADDRESS OF LAST TEST
1359  004374  000137  000200                  JMP     200
1360
1361  004400  022626                  XOR:    CMP     (R6)+,(R6)+
1362  004402  012637  000004                  MOV     (R6)+,@#4
1363  004406  000721                          BR      SCOPEB
1364                                  ;PRINT DEVICE ADDRESS AND VECTOR
1365  004410  012777  000240  000450  MOREID: MOV     #240,@$TPB
1366  004416  105777  000442                  TSTB    @$TPS
1367  004422  100375                          BPL     .-4
1368  004424  013702  001216                  MOV     @#DRCSR,R2
1369  004430  004737  003756                  JSR     R7,PRTAB
1370  004434  012777  000240  000424          MOV     #240,@$TPB
1371  004442  105777  000416                  TSTB    @$TPS
1372  004446  100375                          BPL     .-4
1373  004450  013702  001226                  MOV     DRVECA,R2
1374  004454  004737  003756                  JSR     R7,PRTAB
1375  004460  012777  000215  000400          MOV     #215,@$TPB
1376  004466  105777  000372                  TSTB    @$TPS
1377  004472  100375                          BPL     .-4
1378  004474  012777  000212  000364          MOV     #212,@$TPB
1379  004502  105777  000356                  TSTB    @$TPS
1380  004506  100375                          BPL     .-4
1381  004510  005077  000352                  CLR     @$TPB
1382  004514  105777  000344                  TSTB    @$TPS
1383  004520  100375                          BPL     .-4
1384  004522  000207                          RTS     R7              ;BACK TO PRINT
1385
1386                                  ;ENTER HERE FOR POWER FAIL
1387
1388  004524  010046                  PFAIL:  MOV     R0,-(6)         ;SAVE REGISTER OR STACK
1389  004526  010146                          MOV     R1,-(6)         ;WHEN POWERING DOWN
1390  004530  010246                          MOV     R2,-(6)
1391  004532  010346                          MOV     R3,-(6)
1392  004534  010446                          MOV     R4,-(6)
1393  004536  010546                          MOV     R5,-(6)
1394  004540  013746  000024                  MOV     24,-(6)
1395  004544  010637  004560                  MOV     R6,@#SAVR6      ;STORE STACK POSITION
1396  004550  012737  004562  000024          MOV     #RESTAR,@#24
1397  004556  000000                          HALT                    ;HALT ON POWER DOWN NORMAL
1398  004560  000000                  SAVR6:  0                       ;STACK IS SAVED HERE
1399  004562  013706  004560          RESTAR: MOV     SAVR6,R6        ;RESTORE REGISTER OFF STACK
```

```
1400  004566  012637  000024          MOV     (6)+,24         ;WHEN POWERING UP
1401  004572  012605                  MOV     (6)+,R5
1402  004574  012604                  MOV     (6)+,R4
1403  004576  012603                  MOV     (6)+,R3
1404  004600  012602                  MOV     (6)+,R2
1405  004602  012601                  MOV     (6)+,R1
1406  004604  012600                  MOV     (6)+,R0
1407  004606  000137  001414          JMP     @#RSTART
1408
1409                                ;ENTER HERE FOR UNIQUE SELECTION OF DR11C
1410
1411  004612  000000                SPEC0:  HALT            ;PLACE ADDRESS OF DR11-C CONTROL STATUS
1412  004614  013701  177570                MOV     @#SR,R1
1413  004620  004737  004642                JSR     PC,@#SPEC1
1414  004624  000000                        HALT
1415  004626  013701  177570                MOV     @#SR,R1
1416  004632  004737  004672                JSR     PC,@#SPEC2
1417  004636  000137  001366                JMP     @#START
1418
1419  004642  012700  001200          SPEC1:  MOV     #RCSR,R0        ;SET TABLE ADDRESS
1420  004646  010120                  MOV     R1,(R0)+        ;LOAD INTO TABLE STARTING AT RCSR
1421  004650  062701  000002          ADD     #2,R1           ;STEP TO ADDRESS OF DROUTBUF
1422  004654  010120                  MOV     R1,(R0)+        ;LOAD INTO TABLE
1423  004656  062701  000002          ADD     #2,R1           ;STEP TO ADDRESS OF DRINBUF
1424  004662  010120                  MOV     R1,(R0)+        ;LOAD INTO TABLE
1425  004664  005301                  DEC     R1              ;FORM ADDRESS OF DROUTBUF+1
1426  004666  010120                  MOV     R1,(R0)+        ;LOAD INTO TABLE
1427  004670  000207                  RTS     PC
1428
1429  004672  012700  001210          SPEC2:  MOV     #RCSR1,R0
1430  004676  010120                  MOV     R1,(R0)+        ;LOAD INTO TABLE
1431  004700  005721                  TST     (R1)+
1432  004702  010120                  MOV     R1,(R0)+
1433  004704  005721                  TST     (R1)+
1434  004706  010120                  MOV     R1,(R0)+
1435  004710  000207                  RTS     PC
1436
1437
1438
1439  004712  005015  047531  020125  MESS1:  .ASCIZ  <15><12>'YOU ARE ON AN XOR TESTER'<15><12>
1440  004720  051101  020105  047117
1441  004726  040440  020116  047530
1442  004734  020122  042524  052123
1443  004742  051105  005015     000
1444  004747     015  000012          MCRLF:  .ASCIZ  <15><12>
1445  004752  020040  000040          MSPACE: .ASCIZ  /  /
1446                                  .EVEN
1447                                  ;;*****************************************************************
1448
1449                                  .SBTTL  TYPE ROUTINE
1450
1451                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1452                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1453                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1454                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1455                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
```

```
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*      TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
                                        ;*
                                        ;*2) USING A JSR INSTRUCTION
                                        ;*      MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
                                        ;*      JSR     PC,$TYPE                ;CALL TYPE ROUTINE
                                        ;*      MESADDR                         ;FIRST ADRESS OF MESSAGE
                                        
004756  105737  005073          $TYPE:  TSTB    $TPFLG                  ;IS THERE A TERMINAL?
004762  100002                          BPL     1$                      ;BR IF YES
004764  000000                          HALT                            ;HALT HERE IF NO TERMINAL
004766  000407                          BR      3$                      ;LEAVE
004770  010046                  1$:     MOV     R0,-(SP)                ;SAVE R0
004772  017600  000002                  MOV     @2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
004776  112046                  2$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
005000  001005                          BNE     4$                      ;BR IF IT ISN'T THE TERMINATOR
005002  005726                          TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
005004  012600                          MOV     (SP)+,R0                ;RESTORE R0
005006  062716  000002          3$:     ADD     #2,(SP)                 ;ADJUST RETURN PC
005012  000002                          RTI                             ;RETURN
005014  004737  005046          4$:     JSR     PC,7$                   ;GO TYPE THIS CHARACTER
005020  123726  005072          5$:     CMPB    $FILLC,(SP)+            ;IS IT TIME FOR FILLER CHARS.?
005024  001364                          BNE     2$                      ;IF NO GO GET NEXT CHAR.
005026  013746  005070                  MOV     $NULL,-(SP)             ;GET # OF FILLER CHARS. NEEDED
                                                                        ;AND THE NULL CHAR.
005032  105366  000001          6$:     DECB    1(SP)                   ;DOES A NULL NEED TO BE TYPED?
005036  002770                          BLT     5$                      ;BR IF NO--GO POP THE NULL OFF OF STACK
005040  004737  005046                  JSR     PC,7$                   ;GO TYPE A NULL
005044  000772                          BR      6$                      ;LOOP
005046  105777  000012          7$:     TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
005052  100375                          BPL     7$
005054  116677  000002  000004          MOVB    2(SP),@$TPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
005062  000207                          RTS     PC
005064  177564                  $TPS:   177564                          ;TTY PRINTER STATUS REG. ADDRESS
005066  177566                  $TPB:   177566                          ;TTY PRINTER BUFFER REG. ADDRESS
005070     000                  $NULL:  .BYTE   0                       ;CONTAINS NULL CHARACTER FOR FILLS
005071     002                  $FILLS: .BYTE   2                       ;CONTAINS # OF FILLER CHARACTERS REQUIRED
005072     012                  $FILLC: .BYTE   12                      ;INSERT FILL CHARS. AFTER A "LINE FEED"
005073     000                  $TPFLG: .BYTE   0                       ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
                                        ;;**************************************************************

                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
                                        ;*      TYPOS                           ;CALL FOR TYPEOUT
                                        ;*      .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M                       ;M=1 OR 0
                                        ;*                                              ;1=TYPE LEADING ZEROS
                                        ;*                                              ;0=SUPPRESS LEADING ZEROS
```

```
1512                                    ;*
1513                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
1514                                    ;*$TYPOS OR $TYPOC
1515                                    ;*CALL:
1516                                    ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1517                                    ;*      TYPON                           ;CALL FOR TYPEOUT
1518                                    ;*
1519                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1520                                    ;*CALL:
1521                                    ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1522                                    ;*      TYPOC                           ;CALL FOR TYPEOUT
1523
1524  005074  017646  000000            $TYPOS: MOV     @(SP),-(SP)             ;PICKUP THE MODE
1525  005100  116637  000001  005317            MOVB    1(SP),$OFILL            ;LOAD ZERO FILL SWITCH
1526  005106  112637  005321                    MOVB    (SP)+,$OMODE+1          ;NUMBER OF DIGITS TO TYPE
1527  005112  062716  000002                    ADD     #2,(SP)                 ;ADJUST RETURN ADDRESS
1528  005116  000406                            BR      $TYPON
1529  005120  112737  000001  005317  $TYPOC:  MOVB    #1,$OFILL               ;SET THE ZERO FILL SWITCH
1530  005126  112737  000006  005321            MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
1531  005134  112737  000005  005316  $TYPON:  MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
1532  005142  010346                            MOV     R3,-(SP)                ;SAVE R3
1533  005144  010446                            MOV     R4,-(SP)                ;SAVE R4
1534  005146  010546                            MOV     R5,-(SP)                ;SAVE R5
1535  005150  113704  005321                    MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
1536  005154  005404                            NEG     R4
1537  005156  062704  000006                    ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
1538  005162  110437  005320                    MOVB    R4,$OMODE               ;SAVE IT FOR USE
1539  005166  113704  005317                    MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
1540  005172  016605  000012                    MOV     12(SP),R5               ;PICKUP THE INPUT NUMBER
1541  005176  005003                            CLR     R3                      ;CLEAR THE OUTPUT WORD
1542  005200  006105                    1$:     ROL     R5                      ;ROTATE MSB INTO "C"
1543  005202  000404                            BR      3$                      ;GO DO MSB
1544  005204  006105                    2$:     ROL     R5                      ;FORM THIS DIGIT
1545  005206  006105                            ROL     R5
1546  005210  006105                            ROL     R5
1547  005212  010503                            MOV     R5,R3
1548  005214  006103                    3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
1549  005216  105337  005320                    DECB    $OMODE                  ;TYPE THIS DIGIT?
1550  005222  100016                            BPL     7$                      ;BR IF NO
1551  005224  042703  177770                    BIC     #177770,R3              ;GET RID OF JUNK
1552  005230  001002                            BNE     4$                      ;TEST FOR 0
1553  005232  005704                            TST     R4                      ;SUPPRESS THIS 0?
1554  005234  001403                            BEQ     5$                      ;BR IF YES
1555  005236  005204                    4$:     INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
1556  005240  052703  000060                    BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
1557  005244  052703  000040            5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
1558  005250  110337  005314                    MOVB    R3,8$                   ;SAVE FOR TYPING
1559  005254  104400  005314                    TYPE    ,8$                     ;GO TYPE THIS DIGIT
1560  005260  105337  005316            7$:     DECB    $OCNT                   ;COUNT BY 1
1561  005264  003347                            BGT     2$                      ;BR IF MORE TO DO
1562  005266  002402                            BLT     6$                      ;BR IF DONE
1563  005270  005204                            INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
1564  005272  000744                            BR      2$                      ;GO DO THE LAST DIGIT
1565  005274  012605                    6$:     MOV     (SP)+,R5                ;RESTORE R5
1566  005276  012604                            MOV     (SP)+,R4                ;RESTORE R4
1567  005300  012603                            MOV     (SP)+,R3                ;RESTORE R3
```

```
1568  005302  016666  000002  000004          MOV     2(SP),4(SP)             ;SET THE STACK FOR RETURNING
1569  005310  012616                          MOV     (SP)+,(SP)
1570  005312  000002                          RTI                             ;RETURN
1571  005314     000  
1572  005315     000  
1573  005316     000  
1574  005317     000  
1575  005320  000000
1576
1577
1578                                          .SBTTL  POWER DOWN AND UP ROUTINES
1579
1580                                          ;POWER DOWN ROUTINE
1581  005322  012737  005444  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC        ;SET FOR FAST UP
1582  005330  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;PRIO:7
1583  005336  010046                          MOV     R0,-(SP)                ;PUSH R0 ON STACK
1584  005340  010146                          MOV     R1,-(SP)                ;PUSH R1 ON STACK
1585  005342  010246                          MOV     R2,-(SP)                ;PUSH R2 ON STACK
1586  005344  010346                          MOV     R3,-(SP)                ;PUSH R3 ON STACK
1587  005346  010446                          MOV     R4,-(SP)                ;PUSH R4 ON STACK
1588  005350  010546                          MOV     R5,-(SP)                ;PUSH R5 ON STACK
1589  005352  010637  005450                  MOV     SP,$SAVR6               ;SAVE SP
1590  005356  012737  005370  000024          MOV     #$PWRUP,@#PWRVEC        ;SET UP VECTOR
1591  005364  000000                          HALT
1592  005366  000776                          BR      .-2                     ;HANG UP
1593
1594                                          ;POWER UP ROUTINE
1595  005370  013706  005450          $PWRUP: MOV     $SAVR6,SP               ;GET SP
1596  005374  005037  005450                  CLR     $SAVR6                  ;WAIT LOOP FOR THE TTY
1597  005400  005237  005450          1$:     INC     $SAVR6                  ;WAIT FOR THE INC
1598  005404  001375                          BNE     1$                      ;OF  WORD
1599  005406  012605                          MOV     (SP)+,R5                ;POP STACK INTO R5
1600  005410  012604                          MOV     (SP)+,R4                ;POP STACK INTO R4
1601  005412  012603                          MOV     (SP)+,R3                ;POP STACK INTO R3
1602  005414  012602                          MOV     (SP)+,R2                ;POP STACK INTO R2
1603  005416  012601                          MOV     (SP)+,R1                ;POP STACK INTO R1
1604  005420  012600                          MOV     (SP)+,R0                ;POP STACK INTO R0
1605  005422  012737  005322  000024          MOV     #$PWRDN,@#PWRVEC        ;SET UP THE POWER DOWN VECTOR
1606  005430  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;PRIO:7
1607  005436  104400  005452                  TYPE    ,$POWER                 ;POWER FAIL MESSAGE
1608  005442  000002                          RTI
1609  005444  000000          $ILLUP: HALT                                    ;THE POWER UP SEQUENCE WAS STARTED
1610  005446  000776                          BR      .-2                     ; BEFORE THE POWER DOWN WAS COMPLETE
1611  005450  000000          $SAVR6: 0                                       ;PUT THE SP HERE
1612  005452  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
1613  005460  000122
1614                                          .EVEN
1615
1616
1617                                          .SBTTL  TRAP DECODER
1618
1619                                          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
1620                                          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
1621                                          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
1622                                          ;*GO TO THAT ROUTINE.
1623
```

```
1624  005462  010046                  $TRAP:  MOV     R0,-(SP)                ;SAVE R0
1625  005464  016600  000002                  MOV     2(SP),R0                ;GET TRAP ADDRESS
1626  005470  005740                          TST     -(R0)                   ;BACKUP BY 2
1627  005472  111000                          MOVB    (R0),R0                 ;GET RIGHT BYTE OF TRAP
1628  005474  016000  005502                  MOV     $TRPAD(R0),R0           ;INDEX TO TABLE
1629  005500  000200                          RTS     R0                      ;GO TO ROUTINE
1630
1631
1632                                  .SBTTL  TRAP TABLE
1633
1634                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1635                                  ;*BY THE "TRAP" INSTRUCTION.
1636
1637                                  ;       ROUTINE
1638                                  ;       -------
1639  005502                          $TRPAD:
1640  005502  004756                          $TYPE           ;CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
1641  005504  005120                          $TYPOC          ;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING
1642  005506  005074                          $TYPOS          ;CALL=TYPOS     TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZE
1643  005510  005134                          $TYPON          ;CALL=TYPON     TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST C
1644
1645          017400                          .=017400
1646
1647  017400  000000                  SAVPC:  .WORD 0
1648
1649  017402  000000                  SAVCC:  .WORD 0
1650
1651          000001                          .END
```

```
ASCNT   004236          1281*   1305*   1312*   1316*   1327#
BEGIN   001446           959     964     866#   1358
BINCT   004234          1278*   1297*   1300*   1301    1304*   1326#
BIT0  = 000001           502#
BIT00 = 000001           492#    502
BIT01 = 000002           491#    501
BIT02 = 000004           490#    500
BIT03 = 000010           489#    499
BIT04 = 000020           488#    498
BIT05 = 000040           487#    497
BIT06 = 000100           486#    496
BIT07 = 000200           485#    495
BIT08 = 000400           484#    494
BIT09 = 001000           483#    493
BIT1  = 000002           501#
BIT10 = 002000           482#
BIT11 = 004000           481#
BIT12 = 010000           480#
BIT13 = 020000           479#
BIT14 = 040000           478#
BIT15 = 100000           477#
BIT2  = 000004           500#
BIT3  = 000010           499#
BIT4  = 000020           498#
BIT5  = 000040           497#
BIT6  = 000100           496#
BIT7  = 000200           495#
BIT8  = 000400           494#
BIT9  = 001000           493#
BPTVEC= 000014           509#
BUFTST  001702           914#
COUNT   001236           823#
CSR   = 167770           398#    803     804     805     806
DECML   004230          1283*   1296    1302*   1324#
DISPLA= 177570           434#
DRBHIO  001224           816#    942*    952*    953*
DRCSR   001216           813#    850     866    1101*   1102*   1112*   1113*   1122*   1123*   1133*   1134*   1144*   1145*
                        1147*   1156*   1157*   1159*   1169*   1170*   1172*   1182*   1183*   1184*   1215    1260    1368
DRIBUF  001222           815#    892    1032    1038    1047    1059*   1060
DRLVL   001230           819#   1083    1097*   1190
DROBUF  001220           814#    870*    872     882*    883     890*    898*    899     905*    906     933*    934*    935
                         941*    943     951*    955    1031*   1032*   1037*   1038*   1039    1046*   1047*   1048    1053*
                        1058*   1059
DRVECA  001226           818#   1077*   1083*   1099*   1110*   1120*   1131*   1142*   1154*   1167*   1180*   1190*   1191*
                        1265    1373
DRVECB  001232           820#   1089*
EMTVEC= 000030           512#
END     003406          1194#
END1    003622          1233    1239#
EOPHLT  000250           793#   1203
EPRVEC= 000004           505#
FIRST   001272           827     830     832#
GNS   = ****** U        1640    1641    1642    1643
HDFHM   004214          1317    1319#
HLT   = 104000           396#    874     876     886     894     902     909     924     938     946     957     967     972
                         980     986     993    1001    1005    1013    1021    1027    1034    1041    1050    1063    1071
```

```
                    1081    1093    1104    1115    1125    1136    1149    1161    1174    1187
ICOUNT  004366       869*    881*    889*    897*    913*    975*   1016*   1044*   1056*   1349    1356*
IOTVEC= 000020       510*
LIST    004240      1280    1328*
LOGIC   003610       787    1234*
MCRLF   004747      1249    1269    1444*
MESS1   004712       842    1439*
MINUS   004036      1287    1290*
MKNUM   004066      1296*   1299
MOADD   004150      1308*   1310
MOREID  004410       865    1365*
MSPACE  004752      1254    1259    1264    1445*
PC      =%000007     408*    446*    798*    827*    830*    838*    840*    843*    847*    865*   1203*   1234*   1413*
                    1416*   1427*   1435*   1481*   1488*   1493*
PFAIL   004524       780    1388*
PIRQ    =%177772     432*
PIRQVE= 000240       516*
PL      001240       824*   1097
PRTAB   003756      1278*   1369    1374
PS      =%177776     429*    430
PSW     =%177776     395*    430*    867*    997*   1008*   1075*   1079*   1088*   1091*   1098*   1108*   1109*   1118*
                    1119*   1129*   1130*   1140*   1141*   1153*   1165*   1166*   1178*   1179*   1335*
PWRVEC= 000024       511*   1581*   1582*   1590*   1605*   1606*
RCSR    001200       803*    849    1419
RCSR1   001210       807*   1429
RESTAR  004562      1396    1399*
RESVEC= 000010       506*
RETURN  004372       859*   1336    1354*   1358*
RSTART  001414       858*   1209    1223    1407
R0      =%000000     401*    437*    849*    851     852     853     854     855     856     857     872*    883*    884
                     899*    900     906*    907     918*    920*    922     935*    936     943*    944     965*    969*
                     970     978*   1045*   1046    1048    1051*   1060*   1061    1068*   1069    1211*   1213    1216
                    1217    1218    1219    1220    1221    1222    1232*   1234    1388    1406*   1419*   1420*   1422*
                    1424*   1426*   1429*   1430*   1432*   1434*   1473    1474*   1475    1478*   1583    1604*   1624
                    1625*   1626    1627*   1628*   1629*
R1      =%000001     402*    438*    837*    839*    850*    851*    852*    853*    854*    855*    856*    857*   1389
                    1405*   1412*   1415*   1420    1421*   1422    1423*   1424    1425*   1426    1430    1431   1432
                    1433    1434    1584    1603*
R2      =%000002     403*    439*    914*    915     919*    921*    922    1194    1196*   1204    1205*   1286   1292
                    1368*   1373*   1390    1404*   1585    1602*
R3      =%000003     404*    440*    916*    920     921     927*   1195    1197*   1199*   1206*   1291*   1294*   1298
                    1307*   1309*   1391    1403*   1532    1541*   1547*   1548*   1551*   1556*   1557*   1558    1567*
                    1586    1601*
R4      =%000004     405*    441*    917*    925*    950*    954*    955     958    1280*   1392    1402*   1533    1535*
                    1536*   1537*   1538    1539*   1553    1555*   1563*   1566*   1587    1600*
R5      =%000005     406*    442*    866*    914     964*    965     968*    969     976*    978     983*    984     987*
                     990*    991     994*    998*    999    1002*   1003    1009*   1011    1017*   1018*   1019    1024*
                    1025    1067*   1068    1078*   1082*   1090*   1094*   1213*   1214    1215*   1216*   1217*   1218*
                    1219*   1220*   1221*   1222*   1393    1401*   1534    1540*   1542*   1544*   1545*   1546*   1547
                    1565*   1588    1599*
R6      =%000006     407     443*    445     826*    829*    832*    835     844     845     858*   1076*   1087*   1100*
                    1111*   1121*   1132*   1143*   1155*   1168*   1181*   1334    1343*   1346    1354    1361    1362
                    1395    1399*
R7      =%000007     408     444*    446    1321*   1369*   1374*   1384*
SAVCC   017402      1247*   1255    1649*
SAVPC   017400      1246*   1250    1647*
```

```
SAVR2   003750          1194*   1205    1274#
SAVR3   003752          1195*   1206    1275#
SAVR4   003754          1276#
SAVR6   004560          1395*   1398#   1399
SCOPEA  004360          1348    1354#
SCOPEB  004252          1334#   1340    1352    1363
SCOPEF  004370           860*   1349    1351*   1353*   1357#
SCOPEG  004354          1350    1353#
SEVEN   004226          1282*   1291    1302    1308*   1323#
SP    =%000006           407#    445#    877#   1250*   1255*   1260*   1265*   1473*   1474    1475*   1477    1478    1479*
                        1482    1484*   1486*   1492    1524*   1525    1526    1527*   1532*   1533*   1534*   1540    1565
                        1566    1567    1568*   1569*   1583*   1584*   1585*   1586*   1587*   1588*   1589    1595*   1599
                        1600    1601    1602    1603    1604    1624*   1625
SPEC    001256           790     829#
SPEC0   004612           831    1411#
SPEC1   004642           838    1413    1419#
SPEC2   004672           840    1416    1429#
SR    = 177570           397#    863    1207    1224    1243    1270    1339    1347    1412    1415
STACK = 001100           426#
STAR    004044          1289    1291#   1311
START   001366           791     828     849#   1239    1417
START1  001242           789     826#
STKLMT= 177774           431#
STKPTR= 001200           399#    826     829     858     877    1076    1087    1100    1111    1121    1132    1143    1155
                        1168    1181
SWR   = 177570           433#    434    1201
SW0   = 000001           474#
SW00  = 000001           464#    474
SW01  = 000002           463#    473
SW02  = 000004           462#    472
SW03  = 000010           461#    471
SW04  = 000020           460#    470
SW05  = 000040           459#    469
SW06  = 000100           458#    468    1201
SW07  = 000200           457#    467
SW08  = 000400           456#    466
SW09  = 001000           455#    465
SW1   = 000002           473#
SW10  = 002000           413#    454#   1207
SW11  = 004000           414#    453#   1347
SW12  = 010000           452#
SW13  = 020000           415#    451#   1243
SW14  = 040000           416#    450#   1339
SW15  = 100000           449#
SW2   = 000004           472#
SW3   = 000010           471#
SW4   = 000020           470#
SW5   = 000040           469#
SW6   = 000100           468#
SW7   = 000200           467#
SW8   = 000400           466#
SW9   = 001000           412#    465#    863    1224
TAG     001746           933#
TBITVE= 000014           507#
TINT1   002710          1099    1105#
TINT2   002760          1110    1117#
```

DDDRAA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TINT3   003030            1120    1127#
TINT4   003100            1131    1137#
TINT5   003160            1142    1150#
TINT6   003230            1154    1161#
TINT7   003310            1167    1174#
TINT8   003370            1180    1187#
TKVEC = 000060             514#
TOODLE  004224            1292#   1293#   1294    1322#
TPVEC = 000064             515#
TRAPVE= 000034             513#
TRTVEC= 000014             508#
TST4    002570            1077    1082#
TST5    002640            1089    1094#
TST6    002416            1046#   1054
TST9    002452            1052    1055#
TYPE  = 104400             841     1204    1249    1254    1259    1264    1269    1559    1607    1640#
TYPOC = 104402            1641#
TYPON = 104406            1643#
TYPOS = 104404            1251    1256    1261    1266    1642#
WAIT1   004014            1284#   1285
WAIT2   004172            1313#   1314    1316
WGTCT   004232            1279#   1296#   1298    1303#   1325#
WRTOC   004106            1295    1300#
XLIST   004164            1306    1312#
XOR     004400            1344    1361#
XORA    001352             833     844#
XORFLG  001234             821#    836#    846#   1341
$FILLC  005072            1482    1498#
$FILLS  005071            1497#
$ILLUP  005444            1581    1609#
$NULL   005070            1484    1496#
$OCNT   005316            1531#   1560#   1573#
$OMODE  005320            1526#   1530#   1535    1538#   1549#   1575#
$POWER  005452            1607    1612#
$PWRDN  005322            1581#   1605
$PWRUP  005370            1590    1595#
$RDCHR= ****** U          1644
$RDDEC= ****** U          1644
$RDLIN= ****** U          1644
$RDOCT= ****** U          1644
$SAVRE= ****** U          1644
$SAVR6  005450            1589#   1595    1596#   1597#   1611#
$SETUP= 000014             519#
$STUP = 177777             519#
$SWR  = 000000             518#
$TN   = 000000             517#
$TPB    005066            1226#   1229#   1288#   1290#   1315#   1365#   1370#   1375#   1378#   1381#   1492#   1495#
$TPFLG  005073            1469    1499#
$TPS    005064            1227    1230    1284    1313    1319    1366    1371    1376    1379    1382    1490    1494#
$TRAP   005462             784    1624#
$TRP  = 000010            1631#   1641#   1642#   1643#   1644#
$TRPAD  005502            1628    1639#
$TYPBN= ****** U          1644
$TYPOS= ****** U          1644
$TYPE   004756            1469#   1631    1640
$TYPOC  005120            1529#   1641
```

```
$TYPON  005134          1528    1531#   1643
$TYPOS  005074          1524#   1642
$OFILL  005317          1525*   1529*   1539    1574#
.       = 017404         520#    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     605     607     609     611     613     615     617     619     621
                         623     625     627     629     631     633     635     637     639     641     643     645     647
                         649     651     653     655     657     659     661     663     665     667     669     671     673
                         675     677     679     681     683     685     687     689     691     693     695     697     699
                         701     703     705     707     709     711     713     715     717     719     721     723     725
                         727     729     731     733     735     737     739     741     743     745     747     749     751
                         753     755     757     759     761     763     765     767     769     771     773     775     777#
                         786#    788#    792#    799#    885     893     901     908     923     937     945     956     966
                         971     979     985     992     1000    1004    1012    1020    1026    1033    1040    1049    1062
                        1070    1160    1173    1186    1244    1271    1320    1367    1372    1377    1380    1383    1494
                        1495    1496    1497    1498    1499    1592    1610    1645*
.HLT    003626           782    1243%
.SCOPE  004264           778    1339#
```

DDDRAA.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
COMMEN      1#    517#
ENDCOM      1#    517#
ERROR     427#
ESCAPE      1#    517#
MULT        1#    517#
NEWTST      1#    517#
POP         1#    517#   1599
PUSH        1#    517#   1583
SCOPE     428#    880    888    896    904    912    932    940    948    962    974    982    989    996   1007
         1015   1023   1030   1036   1043   1055   1066   1074   1086   1105   1117   1127   1137   1150   1162
         1175   1188
SETTRA   1631#   1641   1642   1643
SETUP       1#    421#   517#
SKIP        1#    517#
SLASH       1#    517#
SPACE     517#
STARS       1#    517#   1447   1500   1576   1615
TRMTRP   1631#
TYPBIN      1#    517#
TYPDEC      1#    517#
TYPNUM      1#    517#
TYPOCS      1#    517#   1250   1255   1260   1265
TYPOCT      1#    517#
TYPTXT      1#    517#
$$ESCA      1#    517#
$$NEWT      1#    517#
$$SET    1631#   1641   1642   1643
$$SKIP      1#    517#
.EQUAT      1#    420#    422
.HEADE      1#
.KT11       1#
.SETUP      1#    421#    519
.SWRHI      1#
.$CATC      1#    421#
.$CMTA      1#
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
.$POWE      1#    420#   1576
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    420#   1615
.$TYPB      1#
.$TYPD      1#
```

.$TYPE	1#	420#	1447
.$TYPO	1#	420#	1500

DDDRAA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADD      920    921   1220   1221   1222   1296   1300   1302   1308   1421   1423   1479   1527   1537
ASL      927
BEQ      973    885    893    901    908    923    937    945    956    959    966    971    979    985    992
        1000   1004   1012   1033   1040   1049   1052   1062   1070   1202   1208   1233   1244   1295   1305
        1317   1350   1554
BGT     1561
BIC     1002   1091   1098   1101   1108   1112   1118   1122   1129   1133   1140   1144   1147   1156   1159
        1165   1169   1172   1178   1182   1184   1294   1551
BIS      983    990    997    998   1009   1075   1088   1102   1109   1113   1119   1123   1130   1134   1141
        1145   1153   1157   1166   1170   1179   1183   1556   1557
BIT      863   1201   1207   1224   1243   1339   1347
BLT     1487   1562
BMI     1020   1026   1287
BNE      864    926    928   1200   1225   1299   1310   1340   1348   1476   1483   1552   1598
BPL     1228   1231   1271   1285   1314   1320   1342   1367   1372   1377   1380   1383   1470   1491   1550
BR       875    960   1054   1160   1173   1186   1289   1311   1318   1352   1363   1472   1489   1528   1543
        1564   1592   1610
CLR      846    860    918    919    949    951    964    987    994   1017   1031   1044   1045   1053   1079
        1082   1094   1191   1229   1278   1279   1303   1304   1353   1381   1541   1596
CLRB     934    942    952
CMP      844    884    900    907    922    936    944    955    970    984    991    999   1003   1039   1048
        1061   1069   1248   1298   1334   1349   1361
CMPB    1482
COM     1293
DEC      925   1199   1305   1309   1316   1425
DECB    1486   1549   1560
EMT      427
HALT     522    524    526    528    530    532    534    536    538    540    542    544    546    548    550
         552    554    556    558    560    562    564    566    568    570    572    574    576    578    580
         582    584    586    588    590    592    594    596    598    600    602    604    606    608    610
         612    614    616    618    620    622    624    626    628    630    632    634    636    638    640
         642    644    646    648    650    652    654    656    658    660    662    664    666    668    670
         672    674    676    678    680    682    684    686    688    690    692    694    696    698    700
         702    704    706    708    710    712    714    716    718    720    722    724    726    728    730
         732    734    736    738    740    742    744    746    748    750    752    754    756    758    760
         762    764    766    768    770    772    774    776    793   1272   1397   1411   1414   1471   1591
        1609
INC     1018   1051   1297   1351   1555   1563   1597
INCB     953    954
IOT      428
JMP      789    790    791    828    831   1209   1223   1239   1336   1359   1407   1417
JSR      827    830    838    840    865   1203   1234   1369   1374   1413   1416   1481   1488
MOV      826    829    832    833    834    835    836    837    839    845    849    850    851    852    853
         854    855    856    857    858    859    866    867    868    869    870    872    877    878    881
         882    883    889    890    897    898    899    905    906    913    914    916    917    933    935
         941    943    950    965    968    969    975    976    978   1008   1016   1024   1032   1037   1038
        1046   1047   1056   1058   1059   1060   1067   1068   1076   1077   1078   1083   1087   1089   1090
        1097   1099   1100   1110   1111   1120   1121   1131   1132   1142   1143   1154   1155   1167   1168
        1180   1181   1190   1194   1195   1196   1197   1205   1206   1211   1212   1215   1226   1232   1246
        1247   1250   1255   1260   1265   1280   1281   1282   1283   1288   1290   1291   1292   1301   1307
        1312   1315   1335   1343   1344   1345   1346   1354   1362   1365   1368   1370   1373   1375   1378
        1388   1389   1390   1391   1392   1393   1394   1395   1396   1399   1400   1401   1402   1403   1404
        1405   1406   1412   1415   1419   1420   1422   1424   1426   1429   1430   1432   1434   1473   1474
        1478   1484   1524   1532   1533   1534   1540   1547   1565   1566   1567   1568   1569   1581   1582
        1583   1584   1585   1586   1587   1588   1589   1590   1595   1599   1600   1601   1602   1603   1604
        1605   1606   1624   1625   1628
```

```
.MAIN.  MACY11 27(732)  15-OCT-76  15:53  PAGE 46
DDDRAA.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MOVB    1475  1492  1525  1526  1529  1530  1531  1535  1538  1539  1558  1627
NEG     1536
NOP     1080  1092  1103  1114  1124  1135  1146  1148  1158  1171  1185  1235  1236  1237  1238
RESET    871   891   977  1010  1198
ROL     1542  1544  1545  1546  1548
RTI     1245  1273  1355  1480  1570  1608
RTS      798   843   847  1321  1384  1427  1435  1493  1629
SUB     1213  1216  1217  1218  1219
TRAP    1631  1641  1642  1643
TST      892   915  1011  1025  1214  1270  1286  1341  1431  1433  1477  1553  1626
TSTB     958  1019  1227  1230  1284  1313  1319  1366  1371  1376  1379  1382  1469  1490
.ASCIZ  1439  1444  1445  1612
.BYTE   1252  1253  1257  1258  1262  1263  1267  1268  1496  1497  1498  1499  1571  1572  1573
        1574
.ENABL     1   419
.END    1651
.ENDC    427   503   517   519  1253  1254  1258  1259  1263  1264  1268  1269  1448  1501  1577
        1589  1599  1608  1615  1616  1625  1628  1640  1641  1642  1643  1644
.EQUIV   427   428   430   445   446   465   466   467   468   469   470   471   472   473   474
         493   494   495   496   497   498   499   500   501   502
.EVEN   1446  1614
.IF      425   475   503   519  1252  1253  1257  1258  1262  1263  1267  1268  1447  1500  1576
        1589  1599  1607  1608  1612  1615  1624  1628  1631  1641  1642  1643  1644
.IFF     425  1252  1253  1257  1258  1262  1263  1267  1268  1448  1501  1577  1608  1616  1625
.IIF    1251  1256  1261  1266  1494  1495  1496  1497  1498  1499  1640  1641  1642  1643
.IRP     519  1583  1599
.LIST      1     5   419   517   519  1631  1640  1641  1642  1643  1644
.MACRO     1  1631
.MCALL   420   421   517
.NLIST     1     5   419   517   519  1631  1640  1641  1642  1643  1644
.REM       6
.REPT    521
.SBTTL   423  1449  1502  1578  1617  1632
.WORD    961  1647  1649


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DDDRAA.SEQ/SOL/CRF/PAGNUM/NL:TOC/DS:ERFZ=DDDRAA.SML,DDDRAA.P11
RUN-TIME: 16 20 2 SECONDS
RUN-TIME RATIO: 90/39=2.3
CORE USED:  21K  (41 PAGES)
```