DA11F

BUS WINDOW STATIC TEST

MD-11-DZDAA-A

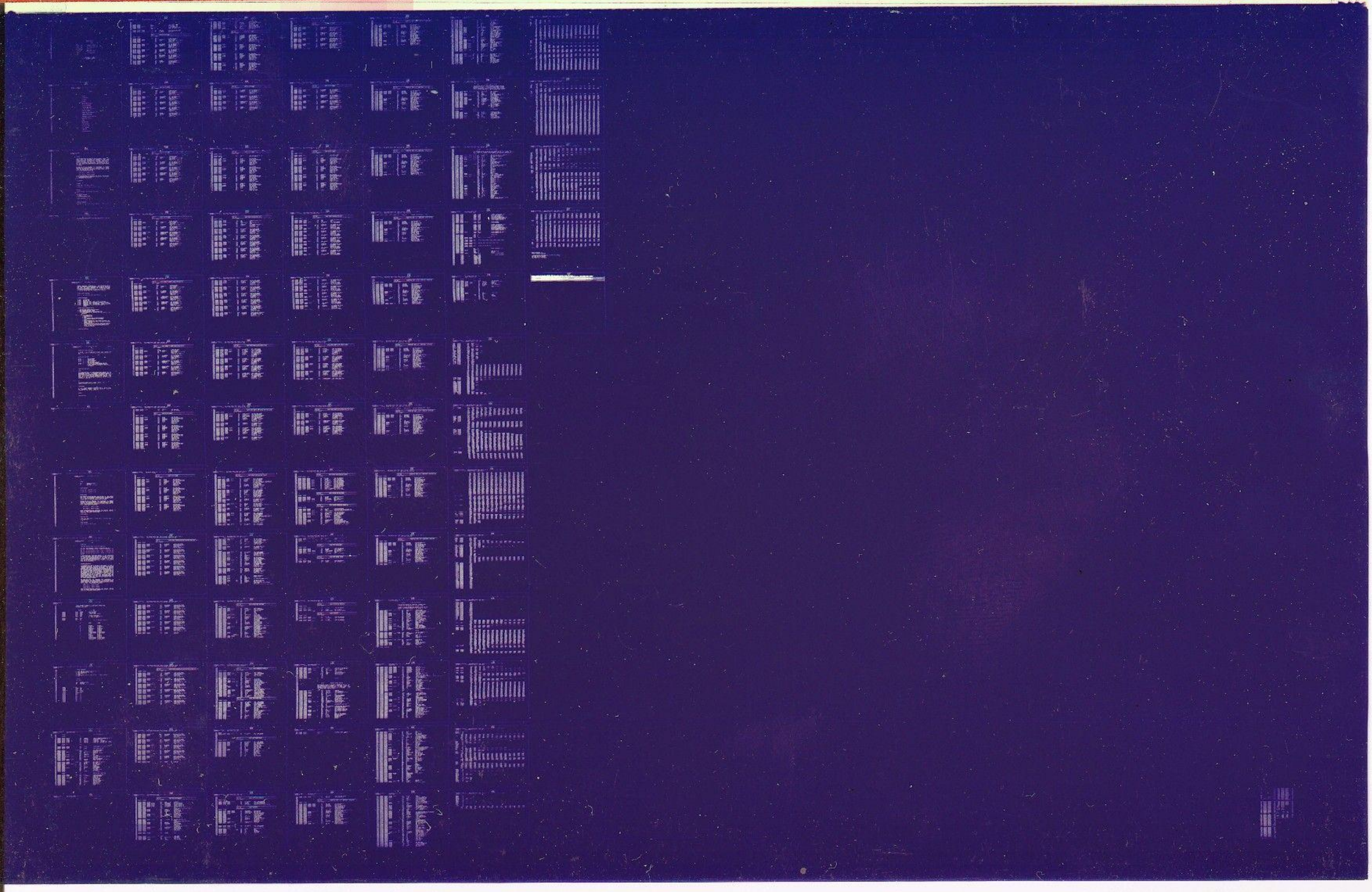
EP-DZDAA-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



```
.REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDAA-A-D
PRODUCT NAME:	BUS WINDOW LOGIC TEST
DATE CREATED:	1-MAY-73
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JOE STUBBLEBINE

CONTENTS

DESCRIPTION

1. ABSTRACT

THIS PROGRAM TESTS THE LOGIC IN THE DA11-F BUS WINDOW. IT FIRST TESTS THE REGISTERS FOR WRITABILITY, THE CROSS COMMUNICATION BITS, AND RELOCATION REGISTERS. THERE ARE TWO DISTINCT SECTIONS. THE FIRST IS A 1 BUS TEST WHICH HAS BOTH SIDES OF THE WINDOW TIED INTO 1 BUS. THE SECOND IS THE 2 BUS TEST WHICH USES HANDSHAKING TO TEST ALL THE REGISTERS. THIS ALSO CHECKS THE WINDOW SIZE AND DOES TRANSFERS THROUGH THE WINDOW.

THE ONE BUS TEST IS PROVIDED FOR CONVENIENCE IN TROUBLE SHOOTING THE DA11-F MODULES. IT IS NOT USED IN THE NORMAL OPERATION OR MAINTENANCE OF A DUAL PROCESSOR SYSTEM. BEFORE USING THE ONE BUS TEST, REMOVE THE FOLLOWING WIRES FROM THE DA11-F WIRED ASSEMBLY BACKPLANE:

BUS A ACLO L: B01F1 TO B04F1

BUS A DCLO L: B01F2 TO B04F2

AFTER TROUBLESHOOTING AN REPAIRING THE MODULES, REPLACE BOTH OF THESE WIRES BEFORE CONNECTING THE DA11-F INTO THE NORMAL DUAL PROCESSOR CONFIGURATION.

2. REQUIREMENTS

2.1 EQUIPMENT

ONE BUS TEST - ANY PDP-11 WITH 4K

TWO BUS TEST - ANY TWO PDP-11'S WITH 4K EACH

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17776

2.3 PRELIMINARY PROGRAMS

THE BASIC PROCESSOR TESTS.

3. LOADING PROCEDURE: SEE 4.3

4. STARTING PROCEDURE

THERE ARE TWO WAYS OF STARTING THE PROGRAM. FIRST OF ALL

```
                                THE  PROGRAM  CAN BE LOADED INTO EACH PROCESSOR BY USING THE
                                ABS LOADER THE PROGRAMS CAN THEN BE STARTED AT THE  STARTING
```

DESCRIPTION

ADDRESS SPECIFIED IN SECTION 4.2 . THE OTHER OPTION IS TO LOAD THE PROGRAM INTO ONE PROCESSOR USING THE ABS LOADER AND THEN HAVE THE WINDOW TRANSFER THE PROGRAM TO THE OTHER PROCESSOR AS DESCRIBED IN SECTION 4.3 . TO USE THIS METHOD THE WINDOW MUST BE WORKING, IT MUST HAVE A SIZE OF AT LEAST 4K, AND THE WINDOW CANNOT BE LOCATED AT 32K OR ABOVE.

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESSES

SA 200 TWO BUS TEST
SA 204 ONE BUS TEST
SA 210 POWER FAIL TEST - HOST - FIRST TO GO DOWN
SA 214 POWER FAIL TEST - NON HOST - SECOND TO GO DOWN
SA 220 POWER FAIL TEST - NEITHER HOST - FIRST TO GO DOWN
SA 224 POWER FAIL TEST - NEITHER HOST - SECOND TO GO DOWN
SA 510 TRANSFERS 0 - 17476 TO OTHER MACHINE (IF > 4K WINDOW)

4.3 PROGRAM AND/OR OPERATOR ACTION

1) LOAD THE ABS LOADER IF NOT IN MEMORY
2) LOAD PROGRAM WITH ABS LOADER
3) LOAD OTHER PROCESSOR IF TWO PROCESSOR TEST.
   A) IF THE WINDOW IS NOT WORKING, USE ABS LOADER ON OTHER PROCESSOR
      OR
   B) IF THE WINDOW IS UP:
      1) LOAD ADDRESS 510
      2) START
      3) LOAD ADDRESS *500 IN OTHER PROCESSOR.
         (* = STARTING ADDRESS OF THE WINDOW)
      4) START
      5) BOTH PROCESSORS WILL SELF START IF SWITCH 7 IS CLEAR. THE HOST WILL DELAY FOR ABOUT 30 SECONDS BEFORE STARTING.
      6) THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

5. OPERATING PROCEDURE

DESCRIPTION

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 .. ALL SWITCHES DOWN IS WORST CASE TESTING FOR THE 2 BUS TEST. THE BELL WILL RING UPON COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE:

```
SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... LOOP ON TEST
SW<13> = 1 ..... INHIBIT PRINTOUT
SW<10> = 1 ..... BELL ON ERROR
           0 ..... BELL ON PASS COMPLETE
SW<08> = 1 ..... TYPE THE TEST NUMBER BEING EXECUTED
SW<07> = 1 ..... IF SET, PREVENTS THE PROGRAM FROM SELF
                 STARTING WHEN THE PROGRAM IS LOADED THROUGH
                 THE WINDOW.
```

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.2.3 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 4 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

6.1 ERROR PRINTOUT

258 THE FORMAT IS AS FOLLOWS:

ADR R0 R1

WHERE:

ADR = ADDRESS OF ERROR HLT
R0 = BAD DATA
R1 = GOOD DATA (OPTIONAL)

6.2 ERROR RECOVERY

ONE BUS TEST - RESTART AT 204

TWO BUS TEST - RESTART AT 200

7. RESTRICTIONS

WHEN RUNNING THE ONE BUS TEST, THE "B" SIDE OF THE WINDOW MUST HAVE ITS CSR REGISTER ADDRESS EXACTLY 20 GREATER THAN THE "A" SIDE CSR REGISTER. IF "A" SIDE CSR ADDRESS EQUALS 764000, THEN "B" SIDE CSR ADDRESS MUST BE 764020.

THE ONE BUS TEST IS PROVIDED FOR CONVENIENCE IN TROUBLE SHOOTING THE DA11-F MODULES. IT IS NOT USED IN THE NORMAL OPERATION OR MAINTENANCE OF A DUAL PROCESSOR SYSTEM. BEFORE USING THE ONE BUS TEST, REMOVE THE FOLLOWING WIRES FROM THE DA11-F WIRED ASSEMBLY BACKPLANE:

BUS A ACLO L: B01F1 TO B04F1

BUS A DCLO L: B01F2 TO B04F2

AFTER TROUBLESHOOTING AND REPAIRING THE MODULES, REPLACE BOTH OF THESE WIRES BEFORE CONNECTING THE DA11-F INTO THE NORMAL DUAL PROCESSOR CONFIGURATION.

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 1 MINUTE WITH ALL SWITCHES DOWN IN THE 2 BUS TEST.

8.2 STACK POINTER

STACK IS INITALLY SET TO 500

DESCRIPTION

8.3 POWER FAIL

THERE ARE SPECIAL ENTRY POINTS FOR POWER FAIL:

SA = 210 HOST COMPUTER - FIRST TO POWER DOWN AND UP
SA = 214 OTHER COMPUTER IS HOST - SECOND TO POWER DOWN AND UP
SA = 220 NEITHER COMPUTER IS HOST - FIRST TO POWER DOWN AND UP
SA = 224 NEITHER COMPUTER IS HOST - SECOND TO POWER DOWN AND UP

TO TEST POWER FAIL, START EACH MACHINE AT ITS APPROPRIATE ADDRESS (SEE ABOVE).THEN TURN THE FIRST MACHINE OFF AND BACK ON. THEN TURN THE SECOND MACHINE OFF AND BACK ON. WHEN FINISHED BOTH MACHINES SHOULD HALT. IF NEITHER MACHINE IS HOST, A THIRD POWER DOEN AND UP IS REQUIRED; I.E., POWER DOWN AND UP THE WINDOW POWER SUPPLY. HOST IS THE SOURCE OF POWER FOR THE BUS WINDOW.

9. PROGRAM DESCRIPTION

THIS PROGRAM IS A TEST OF THE DA11-F BUS WINDOW. IT HAS 2 INDEPENDANT SECTIONS. THE FIRST IS THE 1 BUS TEST WHICH HAS 1 UNIBUS TIED INTO BOTH SIDES OF THE WINDOW. THE REGISTER ADDRESSES AND TRAP VECTORS MUST BE DIFFERENT IN THIS TEST AND TRANSFERS THROUGH THE WINDOW ARE NOT ALLOWED BECAUSE OF THE DATIP CYCLE WOULD CAUSE A TIMEOUT. THIS SHOULD ONLY BE USED FOR INITIAL BRING UP OF THE WINDOW. THE SECOND PART OF THE PROGRAM IS FOR 2 BUSSES AND 2 PROCESSORS. IT USES HANDSHAKING TO CHECK OUT THE REGISTERS. THIS SECTION IS SYNCHRONES AND IS IN 2 SECTIONS. THESE SECTIONS ARE SELECTED DEPENDING UPON WHO GETS THE BUS FIRST AND THEY SWITCH EVERY PASS. IN THIS TEST, TRANSFERS THROUGH THE WINDOW ARE TRIED INCLUDING A SECTION WHICH EXECUTES CODE IN THE OTHER PROCESSOR'S MEMORY.

THE ONE BUS TEST IS PROVIDED FOR CONVENIENCE IN TROUBLESHOOTING THE DA11-F MODULES. IT IS NOT USED IN THE NORMAL OPERATION OR MAINTENANCE OF A DUAL PROCESSOR SYSTEM. BEFORE USING THE ONE BUS TEST, REMOVE THE FOLLOWING WIRES FROM THE DA11-F WIRED ASSEMBLY BACKPLANE:

BUS A ACLO L: B01F1 TO B04F1

BUS A DCLO L: B01F2 TO B04F2

AFTER TROUBLESHOOTING AND REPAIRING THE MODULES, REPLACE BOTH OF THESE WIRES BEFORE CONNECTING THE DA11-F INTO THE NORMAL DUAL PROCESSOR CONFIGURATION.

%

```
                        .TITLE  MAINDEC-11-DZDAA-A      BUS WINDOW STATIC TEST
                        ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                        ;PROGRAM BY BOB BRAIN


                        ;     SWITCH                    USE
                        ;     ------                    ---
100000                  SW15=   100000          ;HALT ON ERROR
040000                  SW14=   40000           ;LOOP ON TEST
020000                  SW13=   20000           ;INHIBIT ERROR TYPEOUTS
010000                  SW12=   10000           ;
004000                  SW11=   4000            ;INHIBIT ITERATIONS
002000                  SW10=   2000            ;0 - BELL ON FASS COMPLETE
                                                ;1 - BELL ON ERROR
001000                  SW9=    1000            ;
000400                  SW8=    400             ;
000200                  SW7=    200                     ;HALT ON TRANSFER COMPLETE


                        ;       BIT             WCSRA           WCSRB
                        ;       ---             -----           -----

                        ;       15              A ERROR         B ERROR
                        ;       14              B TIME OUT      A TIME OUT
                        ;       13              B ACLO          A ACLO
                        ;       12              B NEW DATA      A NEW DATA
                        ;       11              B DATA 3        A DATA 3
                        ;       10              B DATA 2        A DATA 2
                        ;        9              B DATA 1        A DATA 1
                        ;        8              A TRANS ENABLE  B TRANS ENABLE
                        ;        7              B TRANS ENABLE  A TRANS ENABLE
                        ;        6              A I/E           B I/E
                        ;        5              A DATA 3        B DATA 3
                        ;        4              A DATA 2        B DATA 2
                        ;        3              A DATA 1        B DATA 1
                        ;        2              B WRITE ENABLE  A WRITE ENABLE
                        ;        1              A WRITE ENABLE  B WRITE ENABLE
                        ;        0              A NEW DATA      B NEW DATA
```

```
                                ;LOADING PROCEDURE:
                                ;
                                ;1) LOAD THE ABS LOADER IF NOT IN MEMORY
                                ;2) LOAD PROGRAM WITH ABS LOADER
                                ;3) LOAD ADDRESS 510
                                ;4) START
                                ;5) LOAD ADDRESS *500 IN OTHER MACHINE (* = WINDOW ADDRESS)
                                ;6) START
                                ;
                                ;TYPEOUTS:
                                ;
                                ;ADR  R0 R1
                                ;
                                ;ADR    ADDRESS OF ERROR
                                ;R0     BAD DATA
                                ;R1     GOOD DATA IF ANY (OPTIONAL)
                                ;
000001                          N=      1
104400                          SCOPE=  TRAP
104000                          HLT=    EMT
000004                          TYPE=   IOT
177776                          PS=     177776
177570                          SWR=    177570
177570                          DISPLAY=SWR
000007                          BELL=   7
000000                          R0=     %0
000001                          R1=     %1
000002                          R2=     %2
000003                          R3=     %3
000004                          R4=     %4
000005                          R5=     %5
000005                          TTY=    %5
000006                          SP=     %6
000007                          PC=     %7
```

```
                    000000                  .=      0                       ;TRAP CATCHER FROM 0 - 776
                    000200                  .=      200
000200  000137  003410                      JMP     @#BEGIN                 ;JUMP TO BEGINING ADDRESS OF PROGRAM
000204  000137  001014                      JMP     @#ONEBUS                ;START ADDRESS IF ON ONE BUS
000210  000137  010556                      JMP     @#GO.HST                ;POWER FAIL - HOST
000214  000137  011060                      JMP     @#GO.OTH                ;POWER FAIL - NON HOST
000220  000137  011352                      JMP     @#GO.NE1                ;POWER FAIL - NEITHER HOST - FIRST
000224  000137  011776                      JMP     @#GO.NE2                ;POWER FAIL - NEITHER HOST - SECOND
                    000500                  .=      500
000500  000167  000010                      JMP     GETCOR                  ;MOVES POGRAM INTO OTHER MACHINE
000504  000000  000000                      0,0
000510  000137  017440                      JMP     @#OPEN                  ;OPEN WINDOW AND WAIT
000514  010701                  GETCOR:     MOV     PC,R1                   ;GET WINDOW ADDRESS
000516  000005                              RESET                           ;CLEAR THE WORLD
000520  162701  000516                      SUB     #GETCOR+2,R1            ;RELOCATE
000524  005000                              CLR     R0                      ;SETUP FOR TRANSFER
000526  012120                  1$:         MOV     (1)+,(0)+               ;GET IT
000530  022700  017500                      CMP     #17500,R0               ;END?
000534  001374                              BNE     1$                      ;NO - LOOP
000536  032737  000200  177570              BIT     #SW7,@#SWR              ;CHECK FOR HALT ON LOAD COMPLETE
000544  001401                              BEQ     .+4                     ;CONTINUE
000546  000000                              HALT                            ;WAIT FOR CONTINUE
000550  012767  000240  016720              MOV     #NOP,BR.                ;MAKE IT EXECUTE A JMP @#BEGIN
000556  000137  003410                      JMP     @#BEGIN                 ;START THE TEST
000562  012700  164000          REGSCN:     MOV     #164000,R0              ;SET WCSRA FOR SCANNING
000566  012702  000010                      MOV     #8.,R2                  ;SET COUNT
000572  012737  000604  000004              MOV     #5$,@#4                 ;SET FOR TIMEOUT
000600  005710                  6$:         TST     (0)                     ;IS IT THERE?
000602  000207                              RTS     PC                      ;RETURN
000604  022626                  5$:         CMP     (6)+,(6)+               ;CLEAR STACK
000606  062700  000020                      ADD     #20,R0                  ;BUMP ADDRESS
000612  005302                              DEC     R2                      ;DEC COUNT
000614  001371                              BNE     6$                      ;LOOP UNTIL FOUND
000616  000000                              HALT                            ;NO WCSRA
000620  000760                              BR      REGSCN                  ;TRY IT AGAIN
                    000700                  .=      700                     ;EXECUTE FROM OTHER SIDE
000700  012701  000001          CODE1:      MOV     #1,%1                   ;LOAD A BIT FOR SHIFTING
000704  010167  000030                      MOV     %1,CHKADR               ;LOAD THE ADDRESS
000710  000241                  2$:         CLC                             ;CLEAR THE C BIT
000712  006167  000022                      ROL     CHKADR                  ;ROTATE IT
000716  006101                              ROL     %1                      ;GET GOOD DATA
000720  103001                              BCC     .+4                     ;SKIP IF MORE
000722  000207                              RTS     PC                      ;RETURN TO OTHER SIDE
000724  016700  000010                      MOV     CHKADR,%0               ;GET FOR TYPING
000730  020001                              CMP     %0,%1                   ;CHECK FOR BIT
000732  001401                              BEQ     .+4                     ;SKIP IF OK
000734  104001                              HLT     +1                      ;ROTATE FAILED
000736  000764                              BR      2$                      ;LOOP
```

```
   500  000740  000000                CHKADR: 0                       ;ADDRESS OF BIT
```

```
                    001000                  .=      1000

001000  000000      000000                  ICNT:   0                       ;LH = ITERATION COUNT  ;RH = TEST NO.
001002  000000      000000                  ERRORS: 0                       ;ERROR COUNT
001004  000000      000000      000000      PCNT:   0,0                     ;2 WORD PASS COUNT
001010  000000      000000                  LAD:    0                       ;LOOP ADDRESS FOR SCOPE
001012  000000      000000                  HLTADR: 0                       ;LAST HLT INSTRUCTION ADD. EXECUTED

001014  012706      000500      000000      ONEBUS: MOV     #500,SP         ;SET SP ***** 500 *****
001020  004767      011604                          JSR     PC,SETUP        ;SETUP THE VECTORS, ETC.
001024  012737      000340      177776      ONEB1:  MOV     #340,@#PS       ;SET PRIO=7

                                            ;*****************************************************************************
                                            ;TEST 1         CHECK R/W OF WCSRA
                                            ;*****************************************************************************
001032  104400                              TST1:   SCOPE

001034  000005                                      RESET                   ;CLEAR THE WORLD
001036  017700      013762                          MOV     @WCSRA,R0       ;GET THE DATA
001042  022700      000000                          CMP     #0,R0           ;CHECK WCSRA FOR 0 ON RESET
001046  001401                                      BEQ     .+4             ;SKIP IF OK
001050  104000                                      HLT                     ;WCSRA DID NOT RESET TO 0

001052  012777      000573      013744              MOV     #573,@WCSRA     ;SET THE SETABLE BITS
001060  017700      013740                          MOV     @WCSRA,R0       ;GET THE RESULT
001064  022700      000573                          CMP     #573!0!1,R0     ;CHECK FOR PROPER BITS
001070  001401                                      BEQ     .+4             ;SKIP IF OK
001072  104000                                      HLT                     ;WCSRA NOT 573!0

001074  012777      000452      013722              MOV     #452,@WCSRA     ;SET THE SETABLE BITS
001102  017700      013716                          MOV     @WCSRA,R0       ;GET THE RESULT
001106  022700      000453                          CMP     #452!0!1,R0     ;CHECK FOR PROPER BITS
001112  001401                                      BEQ     .+4             ;SKIP IF OK
001114  104000                                      HLT                     ;WCSRA NOT 452!0

001116  012777      000121      013700              MOV     #121,@WCSRA     ;SET THE SETABLE BITS
001124  017700      013674                          MOV     @WCSRA,R0       ;GET THE RESULT
001130  022700      000121                          CMP     #121!0!1,R0     ;CHECK FOR PROPER BITS
001134  001401                                      BEQ     .+4             ;SKIP IF OK
001136  104000                                      HLT                     ;WCSRA NOT 121!0

001140  005077      013660                          CLR     @WCSRA          ;CLEAR ALL SETABLE BITS
001144  017700      013654                          MOV     @WCSRA,R0       ;GET THE RESULT
001150  022700      000001                          CMP     #0!1,R0         ;CHECK FOR PROPER BITS
001154  001401                                      BEQ     .+4             ;SKIP IF OK
001156  104000                                      HLT                     ;WCSRA NOT 0
```

```
                                        ;*****************************************************************************
                                        ;TEST 2          CHECK R/W OF WCSRB
                                        ;*****************************************************************************
001160  104400                  TST2:   SCOPE

001162  000005                          RESET                           ;CLEAR THE WORLD
001164  017700  013662                  MOV     @WCSRB,R0               ;GET THE DATA
001170  022700  000000                  CMP     #0,R0                   ;CHECK WCSRB FOR 0 ON RESET
001174  001401                          BEQ     .+4                     ;SKIP IF OK
001176  104000                          HLT                             ;WCSRB DID NOT RESET TO 0

001200  012777  000573  013644          MOV     #573,@WCSRB             ;SET THE SETABLE BITS
001206  017700  013640                  MOV     @WCSRB,R0               ;GET THE RESULT
001212  022700  000573                  CMP     #573!0!1,R0             ;CHECK FOR PROPER BITS
001216  001401                          BEQ     .+4                     ;SKIP IF OK
001220  104000                          HLT                             ;WCSRB NOT 573!0

001222  012777  000452  013622          MOV     #452,@WCSRB             ;SET THE SETABLE BITS
001230  017700  013616                  MOV     @WCSRB,R0               ;GET THE RESULT
001234  022700  000453                  CMP     #452!0!1,R0             ;CHECK FOR PROPER BITS
001240  001401                          BEQ     .+4                     ;SKIP IF OK
001242  104000                          HLT                             ;WCSRB NOT 452!0

001244  012777  000121  013600          MOV     #121,@WCSRB             ;SET THE SETABLE BITS
001252  017700  013574                  MOV     @WCSRB,R0               ;GET THE RESULT
001256  022700  000121                  CMP     #121!0!1,R0             ;CHECK FOR PROPER BITS
001262  001401                          BEQ     .+4                     ;SKIP IF OK
001264  104000                          HLT                             ;WCSRB NOT 121!0

001266  005077  013560                  CLR     @WCSRB                  ;CLEAR ALL SETABLE BITS
001272  017700  013554                  MOV     @WCSRB,R0               ;GET THE RESULT
001276  022700  000001                  CMP     #0!1,R0                 ;CHECK FOR PROPER BITS
001302  001401                          BEQ     .+4                     ;SKIP IF OK
001304  104000                          HLT                             ;WCSRB NOT 0
```

```
                                        ;*****************************************************************************
                                        ;TEST 3         CHECK R/W OF WDBRA
                                        ;*****************************************************************************
001306  104400                  TST3:   SCOPE

001310  000005                          RESET                           ;CLEAR THE WORLD
001312  017700  013510                  MOV     @WDBRA,R0               ;GET THE DATA
001316  022700  000000                  CMP     #0,R0                   ;CHECK WDBRA FOR 0 ON RESET
001322  001401                          BEQ     .+4                     ;SKIP IF OK
001324  104000                          HLT                             ;WDBRA DID NOT RESET TO 0

001326  012777  177777  013472          MOV     #-1,@WDBRA              ;SET THE SETABLE BITS
001334  017700  013466                  MOV     @WDBRA,R0               ;GET THE RESULT
001340  022700  177777                  CMP     #-1!0,R0                ;CHECK FOR PROPER BITS
001344  001401                          BEQ     .+4                     ;SKIP IF OK
001346  104000                          HLT                             ;WDBRA NOT -1!0

001350  012777  125252  013450          MOV     #125252,@WDBRA          ;SET THE SETABLE BITS
001356  017700  013444                  MOV     @WDBRA,R0               ;GET THE RESULT
001362  022700  125252                  CMP     #125252!0,R0            ;CHECK FOR PROPER BITS
001366  001401                          BEQ     .+4                     ;SKIP IF OK
001370  104000                          HLT                             ;WDBRA NOT 125252!0

001372  012777  052525  013426          MOV     #52525,@WDBRA           ;SET THE SETABLE BITS
001400  017700  013422                  MOV     @WDBRA,R0               ;GET THE RESULT
001404  022700  052525                  CMP     #52525!0,R0             ;CHECK FOR PROPER BITS
001410  001401                          BEQ     .+4                     ;SKIP IF OK
001412  104000                          HLT                             ;WDBRA NOT 52525!0

001414  005077  013406                  CLR     @WDBRA                  ;CLEAR ALL SETABLE BITS
001420  017700  013402                  MOV     @WDBRA,R0               ;GET THE RESULT
001424  022700  000000                  CMP     #0,R0                   ;CHECK FOR PROPER BITS
001430  001401                          BEQ     .+4                     ;SKIP IF OK
001432  104000                          HLT                             ;WDBRA NOT 0
```

```
                                ;*****************************************************************************
                                ;TEST 4          CHECK R/W OF WDBRBX
                                ;*****************************************************************************
001434  104400          TST4:   SCOPE

001436  000005                  RESET                           ;CLEAR THE WORLD
001440  017700  013410          MOV     @WDBRBX,R0              ;GET THE DATA
001444  022700  000000          CMP     #0,R0                   ;CHECK WDBRBX FOR 0 ON RESET
001450  001401                  BEQ     .+4                     ;SKIP IF OK
001452  104000                  HLT                             ;WDBRBX DID NOT RESET TO 0

001454  012777  177777  013372  MOV     #-1,@WDBRBX             ;SET THE SETABLE BITS
001462  017700  013366          MOV     @WDBRBX,R0              ;GET THE RESULT
001466  022700  177777          CMP     #-1!0,R0                ;CHECK FOR PROPER BITS
001472  001401                  BEQ     .+4                     ;SKIP IF OK
001474  104000                  HLT                             ;WDBRBX NOT -1!0

001476  012777  125252  013350  MOV     #125252,@WDBRBX         ;SET THE SETABLE BITS
001504  017700  013344          MOV     @WDBRBX,R0              ;GET THE RESULT
001510  022700  125252          CMP     #125252!0,R0            ;CHECK FOR PROPER BITS
001514  001401                  BEQ     .+4                     ;SKIP IF OK
001516  104000                  HLT                             ;WDBRBX NOT 125252!0

001520  012777  052525  013326  MOV     #52525,@WDBRBX          ;SET THE SETABLE BITS
001526  017700  013322          MOV     @WDBRBX,R0              ;GET THE RESULT
001532  022700  052525          CMP     #52525!0,R0             ;CHECK FOR PROPER BITS
001536  001401                  BEQ     .+4                     ;SKIP IF OK
001540  104000                  HLT                             ;WDBRBX NOT 52525!0

001542  005077  013306          CLR     @WDBRBX                 ;CLEAR ALL SETABLE BITS
001546  017700  013302          MOV     @WDBRBX,R0              ;GET THE RESULT
001552  022700  000000          CMP     #0,R0                   ;CHECK FOR PROPER BITS
001556  001401                  BEQ     .+4                     ;SKIP IF OK
001560  104000                  HLT                             ;WDBRBX NOT 0
```

```
                                ;*****************************************************************************
                                ;TEST 5          CHECK WDBRA TO WDBRAX CROSS COMMUNICATION
                                ;*****************************************************************************
001562  104400                  TST5:   SCOPE

001564  000005                          RESET                   ;CLEAR THE WORLD
001566  017700  013264                  MOV     @WDBRAX,R0      ;GET THE DATA
001572  022700  000000                  CMP     #0,R0           ;CHECK WDBRAX FOR 0 ON RESET
001576  001401                          BEQ     .+4             ;SKIP IF OK
001600  104000                          HLT                     ;WDBRAX DID NOT RESET TO 0

001602  012777  177777  013216          MOV     #-1,@WDBRA      ;SET THE SETABLE BITS
001610  017700  013242                  MOV     @WDBRAX,R0      ;GET THE RESULT
001614  022700  177777                  CMP     #-1!0,R0        ;CHECK FOR PROPER BITS
001620  001401                          BEQ     .+4             ;SKIP IF OK
001622  104000                          HLT                     ;WDBRAX NOT -1!0

001624  012777  125252  013174          MOV     #125252,@WDBRA  ;SET THE SETABLE BITS
001632  017700  013220                  MOV     @WDBRAX,R0      ;GET THE RESULT
001636  022700  125252                  CMP     #125252!0,R0    ;CHECK FOR PROPER BITS
001642  001401                          BEQ     .+4             ;SKIP IF OK
001644  104000                          HLT                     ;WDBRAX NOT 125252!0

001646  012777  052525  013152          MOV     #52525,@WDBRA   ;SET THE SETABLE BITS
001654  017700  013176                  MOV     @WDBRAX,R0      ;GET THE RESULT
001660  022700  052525                  CMP     #52525!0,R0     ;CHECK FOR PROPER BITS
001664  001401                          BEQ     .+4             ;SKIP IF OK
001666  104000                          HLT                     ;WDBRAX NOT 52525!0

001670  005077  013132                  CLR     @WDBRA          ;CLEAR ALL SETABLE BITS
001674  017700  013156                  MOV     @WDBRAX,R0      ;GET THE RESULT
001700  022700  000000                  CMP     #0,R0           ;CHECK FOR PROPER BITS
001704  001401                          BEQ     .+4             ;SKIP IF OK
001706  104000                          HLT                     ;WDBRAX NOT 0
```

```
                                ;*****************************************************************************
                                ;TEST 6          CHECK WDBRBX TO WDBRB CROSS COMMUNICATION
                                ;*****************************************************************************
001710  104400                  TST6:   SCOPE

001712  000005                          RESET                   ;CLEAR THE WORLD
001714  017700  013110                  MOV     @WDBRB,R0       ;GET THE DATA
001720  022700  000000                  CMP     #0,R0           ;CHECK WDBRB FOR 0 ON RESET
001724  001401                          BEQ     .+4             ;SKIP IF OK
001726  104000                          HLT                     ;WDBRB DID NOT RESET TO 0

001730  012777  177777  013116          MOV     #-1,@WDBRBX     ;SET THE SETABLE BITS
001736  017700  013066                  MOV     @WDBRB,R0       ;GET THE RESULT
001742  022700  177777                  CMP     #-1!0,R0        ;CHECK FOR PROPER BITS
001746  001401                          BEQ     .+4             ;SKIP IF OK
001750  104000                          HLT                     ;WDBRB NOT -1!0

001752  012777  125252  013074          MOV     #125252,@WDBRBX ;SET THE SETABLE BITS
001760  017700  013044                  MOV     @WDBRB,R0       ;GET THE RESULT
001764  022700  125252                  CMP     #125252!0,R0    ;CHECK FOR PROPER BITS
001770  001401                          BEQ     .+4             ;SKIP IF OK
001772  104000                          HLT                     ;WDBRB NOT 125252!0

001774  012777  052525  013052          MOV     #52525,@WDBRBX  ;SET THE SETABLE BITS
002002  017700  013022                  MOV     @WDBRB,R0       ;GET THE RESULT
002006  022700  052525                  CMP     #52525!0,R0     ;CHECK FOR PROPER BITS
002012  001401                          BEQ     .+4             ;SKIP IF OK
002014  104000                          HLT                     ;WDBRB NOT 52525!0

002016  005077  013032                  CLR     @WDBRBX         ;CLEAR ALL SETABLE BITS
002022  017700  013002                  MOV     @WDBRB,R0       ;GET THE RESULT
002026  022700  000000                  CMP     #0,R0           ;CHECK FOR PROPER BITS
002032  001401                          BEQ     .+4             ;SKIP IF OK
002034  104000                          HLT                     ;WDBRB NOT 0
```

```
002036  016702  013036                  MOV     SIZE,R2         ;GET PAGE SIZE
002042  005102                          COM     R2              ;GET THE REVERSE

                        ;*************************************************************************
                        ;TEST 7         CHECK R/W OF WRARA
                        ;*************************************************************************
002044  104400          TST7:   SCOPE

002046  012703  000000                  MOV     #0,R3           ;GET THE DATA
002052  040203                          BIC     R2,R3           ;CLEAR THE UNUSED BITS
002054  010377  012754                  MOV     R3,@WRARA       ;SET THE BITS IN WRARA
002060  017700  012750                  MOV     @WRARA,R0       ;GET THE DATA
002064  040200                          BIC     R2,R0           ;CLEAR THE JUNK
002066  020300                          CMP     R3,R0           ;CHECK THE DATA
002070  001401                          BEQ     .+4             ;SKIP IF OK
002072  104000                          HLT                     ;WRARA NOT 0

002074  012703  177777                  MOV     #-1,R3          ;GET THE DATA
002100  040203                          BIC     R2,R3           ;CLEAR THE UNUSED BITS
002102  010377  012726                  MOV     R3,@WRARA       ;SET THE BITS IN WRARA
002106  017700  012722                  MOV     @WRARA,R0       ;GET THE DATA
002112  040200                          BIC     R2,R0           ;CLEAR THE JUNK
002114  020300                          CMP     R3,R0           ;CHECK THE DATA
002116  001401                          BEQ     .+4             ;SKIP IF OK
002120  104000                          HLT                     ;WRARA NOT -1

002122  012703  052525                  MOV     #52525,R3       ;GET THE DATA
002126  040203                          BIC     R2,R3           ;CLEAR THE UNUSED BITS
002130  010377  012700                  MOV     R3,@WRARA       ;SET THE BITS IN WRARA
002134  017700  012674                  MOV     @WRARA,R0       ;GET THE DATA
002140  040200                          BIC     R2,R0           ;CLEAR THE JUNK
002142  020300                          CMP     R3,R0           ;CHECK THE DATA
002144  001401                          BEQ     .+4             ;SKIP IF OK
002146  104000                          HLT                     ;WRARA NOT 52525

002150  012703  125252                  MOV     #125252,R3      ;GET THE DATA
002154  040203                          BIC     R2,R3           ;CLEAR THE UNUSED BITS
002156  010377  012652                  MOV     R3,@WRARA       ;SET THE BITS IN WRARA
002162  017700  012646                  MOV     @WRARA,R0       ;GET THE DATA
002166  040200                          BIC     R2,R0           ;CLEAR THE JUNK
002170  020300                          CMP     R3,R0           ;CHECK THE DATA
002172  001401                          BEQ     .+4             ;SKIP IF OK
002174  104000                          HLT                     ;WRARA NOT 125252
```

```
                                ;*****************************************************************
                                ;TEST 10                    CHECK R/W OF WRARB
                                ;*****************************************************************
002176  104400                  TST10:  SCOPE

002200  012703  000000                  MOV     #0,R3           ;GET THE DATA
002204  040203                          BIC     R2,R3           ;CLEAR THE UNUSED BITS
002206  010377  012650                  MOV     R3,@WRARB       ;SET THE BITS IN WRARB
002212  017700  012644                  MOV     @WRARB,R0       ;GET THE DATA
002216  040200                          BIC     R2,R0           ;CLEAR THE JUNK
002220  020300                          CMP     R3,R0           ;CHECK THE DATA
002222  001401                          BEQ     .+4             ;SKIP IF OK
002224  104000                          HLT                     ;WRARB NOT 0

002226  012703  177777                  MOV     #-1,R3          ;GET THE DATA
002232  040203                          BIC     R2,R3           ;CLEAR THE UNUSED BITS
002234  010377  012622                  MOV     R3,@WRARB       ;SET THE BITS IN WRARB
002240  017700  012616                  MOV     @WRARB,R0       ;GET THE DATA
002244  040200                          BIC     R2,R0           ;CLEAR THE JUNK
002246  020300                          CMP     R3,R0           ;CHECK THE DATA
002250  001401                          BEQ     .+4             ;SKIP IF OK
002252  104000                          HLT                     ;WRARB NOT -1

002254  012703  125252                  MOV     #125252,R3      ;GET     ATA
002260  040203                          BIC     R2,R3           ;CLEA.  :  _ UNUSED BITS
002262  010377  012574                  MOV     R3,@WRARB       ;SET THE BITS IN WRARB
002266  017700  012570                  MOV     @WRARB,R0       ;GET THE DATA
002272  040200                          BIC     R2,R0           ;CLEAR THE JUNK
002274  020300                          CMP     R3,R0           ;CHECK THE DATA
002276  001401                          BEQ     .+4             ;SKIP IF OK
002300  104000                          HLT                     ;WRARB NOT 125252

002302  012703  052525                  MOV     #52525,R3       ;GET THE DATA
002306  040203                          BIC     R2,R3           ;CLEAR THE UNUSED BITS
002310  010377  012546                  MOV     R3,@WRARB       ;SET THE BITS IN WRARB
002314  017700  012542                  MOV     @WRARB,R0       ;GET THE DATA
002320  040200                          BIC     R2,R0           ;CLEAR THE JUNK
002322  020300                          CMP     R3,R0           ;CHECK THE DATA
002324  001401                          BEQ     .+4             ;SKIP IF OK
002326  104000                          HLT                     ;WRARB NOT 52525
```

```
                                ;*****************************************************************
                                ;TEST 11                CHECK CROSS COMMUNICATION BITS BETWEEN WCSRA AND WCSRB
                                ;*****************************************************************
002330  104400                  TST11:  SCOPE

002332  012777  000000  012464          MOV     #0,@WCSRA       ;LOAD 0 INTO WCSRA
002340  017700  012506                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
002344  022700  000000                  CMP     #0,R0           ;CHECK WCSRB FOR 0
002350  001401                          BEQ     .+4             ;SKIP IF OK
002352  104000                          HLT                     ;DATA IN WCSRB NOT 0

002354  012777  000010  012442          MOV     #10,@WCSRA      ;LOAD 10 INTO WCSRA
002362  017700  012464                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
002366  022700  001000                  CMP     #1000,R0        ;CHECK WCSRB FOR 1000
002372  001401                          BEQ     .+4             ;SKIP IF OK
002374  104000                          HLT                     ;DATA IN WCSRB NOT 1000

002376  012777  000020  012420          MOV     #20,@WCSRA      ;LOAD 20 INTO WCSRA
002404  017700  012442                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
002410  022700  002000                  CMP     #2000,R0        ;CHECK WCSRB FOR 2000
002414  001401                          BEQ     .+4             ;SKIP IF OK
002416  104000                          HLT                     ;DATA IN WCSRB NOT 2000

002420  012777  000030  012376          MOV     #30,@WCSRA      ;LOAD 30 INTO WCSRA
002426  017700  012420                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
002432  022700  003000                  CMP     #3000,R0        ;CHECK WCSRB FOR 3000
002436  001401                          BEQ     .+4             ;SKIP IF OK
002440  104000                          HLT                     ;DATA IN WCSRB NOT 3000

002442  012777  000040  012354          MOV     #40,@WCSRA      ;LOAD 40 INTO WCSRA
002450  017700  012376                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
002454  022700  004000                  CMP     #4000,R0        ;CHECK WCSRB FOR 4000
002460  001401                          BEQ     .+4             ;SKIP IF OK
002462  104000                          HLT                     ;DATA IN WCSRB NOT 4000

002464  012777  000050  012332          MOV     #50,@WCSRA      ;LOAD 50 INTO WCSRA
002472  017700  012354                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
002476  022700  005000                  CMP     #5000,R0        ;CHECK WCSRB FOR 5000
002502  001401                          BEQ     .+4             ;SKIP IF OK
002504  104000                          HLT                     ;DATA IN WCSRB NOT 5000
```

```
841
842  002506  012777  000060  012310          MOV     #60,@WCSRA      ;LOAD 60 INTO WCSRA
843  002514  017700  012332                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
844  002520  022700  006000                  CMP     #6000,R0        ;CHECK WCSRB FOR 6000
845  002524  001401                          BEQ     .+4             ;SKIP IF OK
846  002526  104000                          HLT                     ;DATA IN WCSRB NOT 6000
847
848  002530  012777  000070  012266          MOV     #70,@WCSRA      ;LOAD 70 INTO WCSRA
849  002536  017700  012310                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
850  002542  022700  007000                  CMP     #7000,R0        ;CHECK WCSRB FOR 7000
851  002546  001401                          BEQ     .+4             ;SKIP IF OK
852  002550  104000                          HLT                     ;DATA IN WCSRB NOT 7000
853
854  002552  012777  000002  012244          MOV     #2,@WCSRA       ;LOAD 2 INTO WCSRA
855  002560  017700  012266                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
856  002564  022700  000004                  CMP     #4,R0           ;CHECK WCSRB FOR 4
857  002570  001401                          BEQ     .+4             ;SKIP IF OK
858  002572  104000                          HLT                     ;DATA IN WCSRB NOT 4
859
860  002574  012777  000400  012222          MOV     #400,@WCSRA     ;LOAD 400 INTO WCSRA
861  002602  017700  012244                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
862  002606  022700  000200                  CMP     #200,R0         ;CHECK WCSRB FOR 200
863  002612  001401                          BEQ     .+4             ;SKIP IF OK
864  002614  104000                          HLT                     ;DATA IN WCSRB NOT 200
865
866  002616  012777  000001  012200          MOV     #1,@WCSRA       ;LOAD 1 INTO WCSRA
867  002624  017700  012222                  MOV     @WCSRB,R0       ;SAVE DATA FOR TYPING
868  002630  022700  010000                  CMP     #10000,R0       ;CHECK WCSRB FOR 10000
869  002634  001401                          BEQ     .+4             ;SKIP IF OK
870  002636  104000                          HLT                     ;DATA IN WCSRB NOT 10000
871
872  002640  005077  012160                  CLR     @WCSRA          ;CLEAR THE
873  002644  005077  012202                  CLR     @WCSRB          ;REGISTER
```

```
                                ;*****************************************************************
                                ;TEST 12                 CHECK CROSS COMMUNICATION BITS BETWEEN WCSRB AND WCSRA
                                ;*****************************************************************
002650  104400                TST12:  SCOPE

002652  012777  000000  012172          MOV     #0,@WCSRB       ;LOAD 0 INTO WCSRB
002660  017700  012140                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
002664  022700  000000                  CMP     #0,R0           ;CHECK WCSRA FOR 0
002670  001401                          BEQ     .+4             ;SKIP IF OK
002672  104000                          HLT                     ;DATA IN WCSRA NOT 0

002674  012777  000010  012150          MOV     #10,@WCSRB      ;LOAD 10 INTO WCSRB
002702  017700  012116                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
002706  022700  001000                  CMP     #1000,R0        ;CHECK WCSRA FOR 1000
002712  001401                          BEQ     .+4             ;SKIP IF OK
002714  104000                          HLT                     ;DATA IN WCSRA NOT 1000

002716  012777  000020  012126          MOV     #20,@WCSRB      ;LOAD 20 INTO WCSRB
002724  017700  012074                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
002730  022700  002000                  CMP     #2000,R0        ;CHECK WCSRA FOR 2000
002734  001401                          BEQ     .+4             ;SKIP IF OK
002736  104000                          HLT                     ;DATA IN WCSRA NOT 2000

002740  012777  000030  012104          MOV     #30,@WCSRB      ;LOAD 30 INTO WCSRB
002746  017700  012052                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
002752  022700  003000                  CMP     #3000,R0        ;CHECK WCSRA FOR 3000
002756  001401                          BEQ     .+4             ;SKIP IF OK
002760  104000                          HLT                     ;DATA IN WCSRA NOT 3000

002762  012777  000040  012062          MOV     #40,@WCSRB      ;LOAD 40 INTO WCSRB
002770  017700  012030                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
002774  022700  004000                  CMP     #4000,R0        ;CHECK WCSRA FOR 4000
003000  001401                          BEQ     .+4             ;SKIP IF OK
003002  104000                          HLT                     ;DATA IN WCSRA NOT 4000

003004  012777  000050  012040          MOV     #50,@WCSRB      ;LOAD 50 INTO WCSRB
003012  017700  012006                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
003016  022700  005000                  CMP     #5000,R0        ;CHECK WCSRA FOR 5000
003022  001401                          BEQ     .+4             ;SKIP IF OK
003024  104000                          HLT                     ;DATA IN WCSRA NOT 5000
```

```
                914
003026  012777  000060  012016          MOV     #60,@WCSRB      ;LOAD 60 INTO WCSRB
003034  017700  011764                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
003040  022700  006000                  CMP     #6000,R0        ;CHECK WCSRA FOR 6000
003044  001401                          BEQ     .+4             ;SKIP IF OK
003046  104000                          HLT                     ;DATA IN WCSRA NOT 6000
                920
003050  012777  000070  011774          MOV     #70,@WCSRB      ;LOAD 70 INTO WCSRB
003056  017700  011742                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
003062  022700  007000                  CMP     #7000,R0        ;CHECK WCSRA FOR 7000
003066  001401                          BEQ     .+4             ;SKIP IF OK
003070  104000                          HLT                     ;DATA IN WCSRA NOT 7000
                926
003072  012777  000002  011752          MOV     #2,@WCSRB       ;LOAD 2 INTO WCSRB
003100  017700  011720                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
003104  022700  000004                  CMP     #4,R0           ;CHECK WCSRA FOR 4
003110  001401                          BEQ     .+4             ;SKIP IF OK
003112  104000                          HLT                     ;DATA IN WCSRA NOT 4
                932
003114  012777  000400  011730          MOV     #400,@WCSRB     ;LOAD 400 INTO WCSRB
003122  017700  011676                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
003126  022700  000200                  CMP     #200,R0         ;CHECK WCSRA FOR 200
003132  001401                          BEQ     .+4             ;SKIP IF OK
003134  104000                          HLT                     ;DATA IN WCSRA NOT 200
                938
003136  012777  000001  011706          MOV     #1,@WCSRB       ;LOAD 1 INTO WCSRB
003144  017700  011654                  MOV     @WCSRA,R0       ;SAVE DATA FOR TYPING
003150  022700  010000                  CMP     #10000,R0       ;CHECK WCSRA FOR 10000
003154  001401                          BEQ     .+4             ;SKIP IF OK
003156  104000                          HLT                     ;DATA IN WCSRA NOT 10000
                944
003160  005077  011666                  CLR     @WCSRB          ;CLEAR THE
003164  005077  011634                  CLR     @WCSRA          ;REGISTER
```

```
                                ;*****************************************************************
                                ;TEST 13                    CHECK NEW DATA INTERUPT
                                ;*****************************************************************
003170  104400                  TST13:  SCOPE

003172  005037  177776                  CLR     @#PS            ;CLEAR PS
003176  012777  003314  011640          MOV     #1$,@WVECA      ;SET UP A VECTOR
003204  012777  003240  011660          MOV     #2$,@WVECB      ;SET UP B VECTOR
003212  012777  000100  011604          MOV     #100,@WCSRA     ;CLEAR A STATUS I/E
003220  012777  000100  011624          MOV     #100,@WCSRB     ;CLEAR B STATUS I/E
003226  052777  000001  011570          BIS     #1,@WCSRA       ;SET NEW DATA BIT ON A
003234  000240                          NOP                     ;WAIT
003236  104000                          HLT                     ;A SIDE SHOULD HAVE INTERUPTED
003240  012706  000500          2$:     MOV     #500,SP         ;RESET SP
003244  017700  011602                  MOV     @WCSRB,R0       ;GET B STATUS
003250  022700  010100                  CMP     #10100,R0       ;CHECK B STATUS
003254  001401                          BEQ     .+4             ;SKIP IF OK
003256  104000                          HLT                     ;B STATUS IS IS NOT 10100
003260  042777  010000  011564          BIC     #10000,@WCSRB   ;CLEAR THE BIT
003266  017700  011532                  MOV     @WCSRA,R0       ;GET A STATUS
003272  022700  000100                  CMP     #100,R0         ;CHECK A STATUS
003276  001401                          BEQ     .+4             ;SKIP IF OK
003300  104000                          HLT                     ;A STATUS NOT 100

003302  052777  000001  011542          BIS     #1,@WCSRB       ;SET NEW DATA BIT ON B
003310  000240                          NOP                     ;WAIT
003312  104000                          HLT                     ;A SIDE SHOULD HAVE INTERUPTED
003314  012706  000500          1$:     MOV     #500,SP         ;RESET SP
003320  017700  011500                  MOV     @WCSRA,R0       ;GET A STATUS
003324  022700  010100                  CMP     #10100,R0       ;CHECK A STATUS
003330  001401                          BEQ     .+4             ;SKIP IF OK
003332  104000                          HLT                     ;A STATUS IS IS NOT 10100
003334  042777  010000  011462          BIC     #10000,@WCSRA   ;CLEAR THE BIT
003342  017700  011504                  MOV     @WCSRB,R0       ;GET B STATUS
003346  022700  000100                  CMP     #100,R0         ;CHECK B STATUS
003352  001401                          BEQ     .+4             ;SKIP IF OK
003354  104000                          HLT                     ;B STATUS NOT 100

003356  005077  011442                  CLR     @WCSRA          ;I/D
003362  005077  011464                  CLR     @WCSRB          ;I/D


003366  104400                          SCOPE                   ;LAST SCOPE
003370  032737  002000  177570          BIT     #SW10,@#SWR     ;RING BELL?
003376  001002                          BNE     3$              ;SKIP IF SET
003400  000004  000007                  TYPE    ,BELL           ;RING THE BELL
003404  000167  175404          3$:     JMP     ONEBUS          ;LOOP ON ONEBUS TEST
```

```
                                BEGIN:  RESET                   ;CLEAR THE WORLD
 994  003410  000005                    MOV     #500,SP         ;SET SP ***** 500 *****
 995  003412  012706  000500            JSR     PC,SETUP        ;SETUP VECTORS, ETC.
 996  003416  004767  007206            MOV     #340,@#PS       ;SET PRIO=7
 997  003422  012737  000340  177776 GO: NOP                     ;GIVE IT SOME ROOM
 998  003430  000240                    BIT     #10000,@WCSRA   ;CHECK FOR NEW DATA
 999  003432  032777  010000  011364    BEQ     .+6             ;SKIP IF FIRST ON THE BUS
1000  003440  001402                    JMP     SECOND          ;START SECOND SECTION
1001  003442  000167  002752            MOVB    #100,ICNT       ;SET TEST 101
1002  003446  112767  000100  175324
1003
1004                                    ;*****************************************************************************
1005                                    ;TEST 101               CHECK R/W OF WCSRA
1006                                    ;*****************************************************************************
1007  003454  104400                TST101: SCOPE
1008
1009  003456  012777  000001  011340    MOV     #1,@WCSRA       ;CLEAR REGISTER
1010  003464  017702  011334            MOV     @WCSRA,R2       ;GET THE BITS
1011  003470  012777  000001  011326    MOV     #1,@WCSRA       ;SET THE NEW DATA BIT
1012  003476  017700  011322            MOV     @WCSRA,R0       ;GET FOR TYPING
1013  003502  040200                    BIC     R2,R0           ;CLEAR THE JUNK
1014  003504  005700                    TST     R0              ;CHECK FOR 0 BITS (1 MASKED)
1015  003506  001401                    BEQ     .+4             ;SKIP IF OK
1016  003510  104000                    HLT                     ;NEW DATA DID NOT SET
1017
1018
1019  003512  012703  000573            MOV     #573,R3         ;GET THE DATA
1020  003516  040203                    BIC     R2,R3           ;CLEAR THE UNUSED BITS
1021  003520  010377  011300            MOV     R3,@WCSRA       ;SET THE BITS IN WCSRA
1022  003524  017700  011274            MOV     @WCSRA,R0       ;GET THE DATA
1023  003530  040200                    BIC     R2,R0           ;CLEAR THE JUNK
1024  003532  020300                    CMP     R3,R0           ;CHECK THE DATA
1025  003534  001401                    BEQ     .+4             ;SKIP IF OK
1026  003536  104000                    HLT                     ;WCSRA NOT 573
1027
1028
1029  003540  012703  000453            MOV     #453,R3         ;GET THE DATA
1030  003544  040203                    BIC     R2,R3           ;CLEAR THE UNUSED BITS
1031  003546  010377  011252            MOV     R3,@WCSRA       ;SET THE BITS IN WCSRA
1032  003552  017700  011246            MOV     @WCSRA,R0       ;GET THE DATA
1033  003556  040200                    BIC     R2,R0           ;CLEAR THE JUNK
1034  003560  020300                    CMP     R3,R0           ;CHECK THE DATA
1035  003562  001401                    BEQ     .+4             ;SKIP IF OK
1036  003564  104000                    HLT                     ;WCSRA NOT 453
1037
1038
1039  003566  012703  000121            MOV     #121,R3         ;GET THE DATA
1040  003572  040203                    BIC     R2,R3           ;CLEAR THE UNUSED BITS
1041  003574  010377  011224            MOV     R3,@WCSRA       ;SET THE BITS IN WCSRA
1042  003600  017700  011220            MOV     @WCSRA,R0       ;GET THE DATA
1043  003604  040200                    BIC     R2,R0           ;CLEAR THE JUNK
1044  003606  020300                    CMP     R3,R0           ;CHECK THE DATA
1045  003610  001401                    BEQ     .+4             ;SKIP IF OK
1046  003612  104000                    HLT                     ;WCSRA NOT 121
1047  003614  005077  011204            CLR     @WCSRA          ;ZERO THE REGISTER
```

```
1048                                    ;*****************************************************************************
1049                                    ;TEST 102                   CHECK R/W OF WDBRA
1050                                    ;*****************************************************************************
1051  003620  104400                    TST102: SCOPE
1052
1053  003622  012777  177777  011176            MOV     #-1,@WDBRA      ;SET THE SETABLE BITS
1054  003630  017700  011172                    MOV     @WDBRA,R0       ;GET THE RESULT
1055  003634  022700  177777                    CMP     #-1!0,R0        ;CHECK FOR PROPER BITS
1056  003640  001401                            BEQ     .+4             ;SKIP IF OK
1057  003642  104000                            HLT                     ;WDBRA NOT -1!0
1058
1059  003644  012777  052525  011154            MOV     #52525,@WDBRA   ;SET THE SETABLE BITS
1060  003652  017700  011150                    MOV     @WDBRA,R0       ;GET THE RESULT
1061  003656  022700  052525                    CMP     #52525!0,R0     ;CHECK FOR PROPER BITS
1062  003662  001401                            BEQ     .+4             ;SKIP IF OK
1063  003664  104000                            HLT                     ;WDBRA NOT 52525!0
1064
1065  003666  012777  125252  011132            MOV     #125252,@WDBRA  ;SET THE SETABLE BITS
1066  003674  017700  011126                    MOV     @WDBRA,R0       ;GET THE RESULT
1067  003700  022700  125252                    CMP     #125252!0,R0    ;CHECK FOR PROPER BITS
1068  003704  001401                            BEQ     .+4             ;SKIP IF OK
1069  003706  104000                            HLT                     ;WDBRA NOT 125252!0
1070
1071  003710  005077  011112                    CLR     @WDBRA          ;CLEAR ALL SETABLE BITS
1072  003714  017700  011106                    MOV     @WDBRA,R0       ;GET THE RESULT
1073  003720  022700  000000                    CMP     #0,R0           ;CHECK FOR PROPER BITS
1074  003724  001401                            BEQ     .+4             ;SKIP IF OK
1075  003726  104000                            HLT                     ;WDBRA NOT 0
```

```
1076                                    ;*****************************************************************************
1077                                    ;TEST 103                       CHECK R/W OF WRARA
1078                                    ;*****************************************************************************
1079  003730  104400                    TST103: SCOPE
1080
1081  003732  016702  011142                    MOV     SIZE,R2         ;GET THE SIZE OF THE WINDOW
1082  003736  005102                    COM     R2              ;SWITCH THE BITS
1083
1084  003740  012703  000000                    MOV     #0,R3           ;GET THE DATA
1085  003744  040203                    BIC     R2,R3           ;CLEAR THE UNUSED BITS
1086  003746  010377  011062                    MOV     R3,@WRARA       ;SET THE BITS IN WRARA
1087  003752  017700  011056                    MOV     @WRARA,R0       ;GET THE DATA
1088  003756  040200                    BIC     R2,R0           ;CLEAR THE JUNK
1089  003760  020300                    CMP     R3,R0           ;CHECK THE DATA
1090  003762  001401                    BEQ     .+4             ;SKIP IF OK
1091  003764  104000                    HLT                     ;WRARA NOT 0
1092
1093  003766  012703  177777                    MOV     #-1,R3          ;GET THE DATA
1094  003772  040203                    BIC     R2,R3           ;CLEAR THE UNUSED BITS
1095  003774  010377  011034                    MOV     R3,@WRARA       ;SET THE BITS IN WRARA
1096  004000  017700  011030                    MOV     @WRARA,R0       ;GET THE DATA
1097  004004  040200                    BIC     R2,R0           ;CLEAR THE JUNK
1098  004006  020300                    CMP     R3,R0           ;CHECK THE DATA
1099  004010  001401                    BEQ     .+4             ;SKIP IF OK
1100  004012  104000                    HLT                     ;WRARA NOT -1
1101
1102  004014  012703  125252                    MOV     #125252,R3      ;GET THE DATA
1103  004020  040203                    BIC     R2,R3           ;CLEAR THE UNUSED BITS
1104  004022  010377  011006                    MOV     R3,@WRARA       ;SET THE BITS IN WRARA
1105  004026  017700  011002                    MOV     @WRARA,R0       ;GET THE DATA
1106  004032  040200                    BIC     R2,R0           ;CLEAR THE JUNK
1107  004034  020300                    CMP     R3,R0           ;CHECK THE DATA
1108  004036  001401                    BEQ     .+4             ;SKIP IF OK
1109  004040  104000                    HLT                     ;WRARA NOT 125252
1110
1111  004042  012703  052525                    MOV     #52525,R3       ;GET THE DATA
1112  004046  040203                    BIC     R2,R3           ;CLEAR THE UNUSED BITS
1113  004050  010377  010760                    MOV     R3,@WRARA       ;SET THE BITS IN WRARA
1114  004054  017700  010754                    MOV     @WRARA,R0       ;GET THE DATA
1115  004060  040200                    BIC     R2,R0           ;CLEAR THE JUNK
1116  004062  020300                    CMP     R3,R0           ;CHECK THE DATA
1117  004064  001401                    BEQ     .+4             ;SKIP IF OK
1118  004066  104000                    HLT                     ;WRARA NOT 52525
1119  004070  005077  010740                    CLR     @WRARA          ;CLEAR THE JUNK
```

```
1120                                    ;*************************************************************************
1121                                    ;TEST 104                CHECK CROSS COMMUNICATION BITS
1122                                    ;*************************************************************************
1123  004074  104400                    TST104: SCOPE
1124
1125  004076  005037  000006                    CLR     @#6             ;CLEAR THE STATUS WORD
1126  004102  005037  177776                    CLR     @#PS            ;SPL 0
1127  004106  032777  010000  010710            BIT     #10000,@WCSRA   ;WAIT FOR NEW DATA
1128  004114  001774                            BEQ     .-6             ;HANG
1129  004116  017700  010702                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1130  004122  022700  010000                    CMP     #10000,R0       ;IS @WCSRA = 10000?
1131  004126  001401                            BEQ     .+4             ;SKIP IF OK
1132  004130  104000                            HLT                     ;@WCSRA NOT 10000
1133
1134  004132  012777  000001  010664            MOV     #0!1,@WCSRA     ;SET 0 INTO @WCSRA
1135  004140  004767  010214                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1136  004144  017700  010654                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1137  004150  022700  010000                    CMP     #10000,R0       ;IS @WCSRA = 10000?
1138  004154  001401                            BEQ     .+4             ;SKIP IF OK
1139  004156  104000                            HLT                     ;@WCSRA NOT 10000
1140
1141  004160  012777  000011  010636            MOV     #10!1,@WCSRA    ;SET 10 INTO @WCSRA
1142  004166  004767  010166                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
1143  004172  017700  010626                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1144  004176  022700  011010                    CMP     #11010,R0       ;IS @WCSRA = 11010?
1145  004202  001401                            BEQ     .+4             ;SKIP IF OK
1146  004204  104000                            HLT                     ;@WCSRA NOT 11010
1147
1148  004206  012777  000021  010610            MOV     #20!1,@WCSRA    ;SET 20 INTO @WCSRA
1149  004214  004767  010140                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
1150  004220  017700  010600                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1151  004224  022700  012020                    CMP     #12020,R0       ;IS @WCSRA = 12020?
1152  004230  001401                            BEQ     .+4             ;SKIP IF OK
1153  004232  104000                            HLT                     ;@WCSRA NOT 12020
1154
1155  004234  012777  000031  010562            MOV     #30!1,@WCSRA    ;SET 30 INTO @WCSRA
1156  004242  004767  010112                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 4 ..
1157  004246  017700  010552                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1158  004252  022700  013030                    CMP     #13030,R0       ;IS @WCSRA = 13030?
1159  004256  001401                            BEQ     .+4             ;SKIP IF OK
1160  004260  104000                            HLT                     ;@WCSRA NOT 13030
1161
1162  004262  012777  000041  010534            MOV     #40!1,@WCSRA    ;SET 40 INTO @WCSRA
1163  004270  004767  010064                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 5 ..
1164  004274  017700  010524                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1165  004300  022700  014040                    CMP     #14040,R0       ;IS @WCSRA = 14040?
1166  004304  001401                            BEQ     .+4             ;SKIP IF OK
1167  004306  104000                            HLT                     ;@WCSRA NOT 14040
1168
```

```
1169  004310  012777  000051  010506          MOV     #50!1,@WCSRA    ;SET 50 INTO @WCSRA
1170  004316  004767  010036                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 6 ..
1171  004322  017700  010476                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1172  004326  022700  015050                  CMP     #15050,R0       ;IS @WCSRA = 15050?
1173  004332  001401                          BEQ     .+4             ;SKIP IF OK
1174  004334  104000                          HLT                     ;@WCSRA NOT 15050
1175
1176  004336  012777  000061  010460          MOV     #60!1,@WCSRA    ;SET 60 INTO @WCSRA
1177  004344  004767  010010                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 7 ..
1178  004350  017700  010450                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1179  004354  022700  016060                  CMP     #16060,R0       ;IS @WCSRA = 16060?
1180  004360  001401                          BEQ     .+4             ;SKIP IF OK
1181  004362  104000                          HLT                     ;@WCSRA NOT 16060
1182
1183  004364  012777  000071  010432          MOV     #70!1,@WCSRA    ;SET 70 INTO @WCSRA
1184  004372  004767  007762                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 10 ..
1185  004376  017700  010422                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1186  004402  022700  017070                  CMP     #17070,R0       ;IS @WCSRA = 17070?
1187  004406  001401                          BEQ     .+4             ;SKIP IF OK
1188  004410  104000                          HLT                     ;@WCSRA NOT 17070
1189
1190  004412  012777  000003  010404          MOV     #2!1,@WCSRA     ;SET 2 INTO @WCSRA
1191  004420  004767  007734                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 11 ..
1192  004424  017700  010374                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1193  004430  022700  010006                  CMP     #10006,R0       ;IS @WCSRA = 10006?
1194  004434  001401                          BEQ     .+4             ;SKIP IF OK
1195  004436  104000                          HLT                     ;@WCSRA NOT 10006
1196
1197  004440  012777  000401  010356          MOV     #400!1,@WCSRA   ;SET 400 INTO @WCSRA
1198  004446  004767  007706                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 12 ..
1199  004452  017700  010346                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1200  004456  022700  010600                  CMP     #10600,R0       ;IS @WCSRA = 10600?
1201  004462  001401                          BEQ     .+4             ;SKIP IF OK
1202  004464  104000                          HLT                     ;@WCSRA NOT 10600
1203
1204  004466  112777  000000  010330          MOVB    #0,@WCSRA       ;CHECK FOR BYTE ADDRESSABLE
1205  004474  017700  010324                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1206  004500  022700  010600                  CMP     #10600,R0       ;IS @WCSRA = 10600?
1207  004504  001401                          BEQ     .+4             ;SKIP IF OK
1208  004506  104000                          HLT                     ;@WCSRA NOT 10600
```

```
1209                                    ;*************************************************************************
1210                                    ;TEST 105                   CHECK CROSS WDBR'S WITH WINDOW CLOSED
1211                                    ;*************************************************************************
1212  004510  104400                    TST105: SCOPE
1213
1214  004512  012777  000001  010304            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1215  004520  004767  007634                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1216  004524  012777  177777  010274            MOV     #-1,@WDBRA      ;LOAD -1 INTO @WDBRA
1217  004532  012777  000001  010264            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1218  004540  004767  007614                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
1219  004544  017700  010260                    MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
1220  004550  022700  177777                    CMP     #-1,R0          ;IS @WDBRB = -1?
1221  004554  001401                            BEQ     .+4             ;SKIP IF OK
1222  004556  104000                            HLT                     ;@WDBRB NOT -1
1223
1224  004560  012777  000000  010240            MOV     #0,@WDBRA       ;LOAD 0 INTO @WDBRA
1225  004566  012777  000001  010230            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1226  004574  004767  007560                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
1227  004600  017700  010224                    MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
1228  004604  022700  000000                    CMP     #0,R0           ;IS @WDBRB = 0?
1229  004610  001401                            BEQ     .+4             ;SKIP IF OK
1230  004612  104000                            HLT                     ;@WDBRB NOT 0
1231
1232  004614  012777  052525  010204            MOV     #52525,@WDBRA   ;LOAD 52525 INTO @WDBRA
1233  004622  012777  000001  010174            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1234  004630  004767  007524                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 4 ..
1235  004634  017700  010170                    MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
1236  004640  022700  052525                    CMP     #52525,R0       ;IS @WDBRB = 52525?
1237  004644  001401                            BEQ     .+4             ;SKIP IF OK
1238  004646  104000                            HLT                     ;@WDBRB NOT 52525
1239
1240  004650  012777  125252  010150            MOV     #125252,@WDBRA  ;LOAD 125252 INTO @WDBRA
1241  004656  012777  000001  010140            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1242  004664  004767  007470                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 5 ..
1243  004670  017700  010134                    MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
1244  004674  022700  125252                    CMP     #125252,R0      ;IS @WDBRB = 125252?
1245  004700  001401                            BEQ     .+4             ;SKIP IF OK
1246  004702  104000                            HLT                     ;@WDBRB NOT 125252
```

```
                                ;*****************************************************************
                                ;TEST 106                CHECK CROSS WDBR'S WITH WINDOW OPEN AND CLOSED
                                ;*****************************************************************
004704  104400                TST106: SCOPE

004706  012777  000401  010110          MOV     #400!1,@WCSRA   ;SET 400 INTO @WCSRA
004714  004767  007440                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
004720  012777  052525  010100          MOV     #52525,@WDBRA   ;LOAD 52525 INTO @WDBRA
004726  012777  000401  010070          MOV     #400!1,@WCSRA   ;SET 400 INTO @WCSRA
004734  004767  007420                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
004740  017700  010062                  MOV     @WDBRA,R0       ;GET @WDBRA FOR TYPING
004744  022700  125252                  CMP     #125252,R0      ;IS @WDBRA = 125252?
004750  001401                          BEQ     .+4             ;SKIP IF OK
004752  104000                          HLT                     ;@WDBRA NOT 125252

004754  017700  010050                  MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
004760  022700  125252                  CMP     #125252,R0      ;IS @WDBRB = 125252?
004764  001401                          BEQ     .+4             ;SKIP IF OK
004766  104000                          HLT                     ;@WDBRB NOT 125252

004770  012777  000001  010026          MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
004776  004767  007356                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
005002  012777  052525  010016          MOV     #52525,@WDBRA   ;LOAD 52525 INTO @WDBRA
005010  012777  000001  010006          MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
005016  004767  007336                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 4 ..
005022  017700  010000                  MOV     @WDBRA,R0       ;GET @WDBRA FOR TYPING
005026  022700  052525                  CMP     #52525,R0       ;IS @WDBRA = 52525?
005032  001401                          BEQ     .+4             ;SKIP IF OK
005034  104000                          HLT                     ;@WDBRA NOT 52525

005036  017700  007766                  MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
005042  022700  052525                  CMP     #52525,R0       ;IS @WDBRB = 52525?
005046  001401                          BEQ     .+4             ;SKIP IF OK
005050  104000                          HLT                     ;@WDBRB NOT 52525
```

```
1281                                    ;*****************************************************************
1282                                    ;TEST 107               CHECK ERROR CONDITIONS AND LOCKOUT
1283                                    ;*****************************************************************
1284  005052  104400                    TST107: SCOPE
1285
1286  005054  012777  000001  007742            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1287  005062  004767  007272                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1288  005066  012737  005104  000004            MOV     #1$,@#4         ;SET FOR TIMEOUT
1289  005074  005777  007750                    TST     @WADRA          ;TRY TO GO THROUGH A CLOSED WINDOW
1290  005100  104000                            HLT                     ;NO TRAP WITH CLOSED WINDOW (BROKE IT)
1291  005102  000401                            BR      .+4             ;SKIP THE CLEAR
1292  005104  022626                    1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK
1293  005106  017700  007712                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1294  005112  022700  110000                    CMP     #110000,R0      ;IS @WCSRA = 110000?
1295  005116  001401                            BEQ     .+4             ;SKIP IF OK
1296  005120  104000                            HLT                     ;@WCSRA NOT 110000
1297
1298  005122  012777  000001  007674            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1299  005130  004767  007224                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
1300  005134  012737  005150  000004            MOV     #2$,@#4         ;SET FOR TIMEOUT
1301  005142  005777  007702                    TST     @WADRA          ;NOW TRY THE WINDOW
1302  005146  000405                            BR      3$              ;SHOULD NOT TRAP
1303  005150  017700  007650            2$:     MOV     @WCSRA,R0       ;GET FOR TYPING
1304  005154  104000                            HLT                     ;WINDOW TRAPPED ON DATAI
1305  005156  022626                            CMP     (6)+,(6)+       ;CLEAR THE STACK
1306  005160  000406                            BR      4$              ;SKIP THE TEST
1307  005162                            3$:
1308  005162  017700  007636                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1309  005166  022700  010200                    CMP     #10200,R0       ;IS @WCSRA = 10200?
1310  005172  001401                            BEQ     .+4             ;SKIP IF OK
1311  005174  104000                            HLT                     ;@WCSRA NOT 10200
1312
1313  005176  012737  005212  000004    4$:     MOV     #5$,@#4         ;SET FOR TIMEOUT
1314  005204  005077  007640                    CLR     @WADRA          ;TRY DATAO WITH ONLY TRANSFER EN SET
1315  005210  000401                            BR      14$             ;SKIP THE CLEAR
1316  005212  022626                    5$:     CMP     (6)+,(6)+       ;CLEAR THE STACK
1317  005214                            14$:
1318  005214  017700  007604                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1319  005220  022700  110200                    CMP     #110200,R0      ;IS @WCSRA = 110200?
1320  005224  001401                            BEQ     .+4             ;SKIP IF OK
1321  005226  104000                            HLT                     ;@WCSRA NOT 110200
1322
1323  005230  012777  000001  007566            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1324  005236  004767  007116                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
1325  005242  012737  005260  000004            MOV     #6$,@#4         ;SET FOR TIMEOUT
1326  005250  005077  007574                    CLR     @WADRA          ;TRY DATAO WITH ONLY WRITE ONLY SET
1327  005254  104000                            HLT                     ;DID NOT TRAP ON ONLY WRITE ONLY
1328  005256  000401                            BR      .+4             ;SKIP CLEAR
1329  005260  022626                    6$:     CMP     (6)+,(6)+       ;CLEAR THE JUNK
1330  005262  017700  007536                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1331  005266  022700  110004                    CMP     #110004,R0      ;IS @WCSRA = 110004?
1332  005272  001401                            BEQ     .+4             ;SKIP IF OK
1333  005274  104000                            HLT                     ;@WCSRA NOT 110004
1334
1335  005276  012777  000001  007520            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1336  005304  004767  007050                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 4 ..
```

```
1337  005310  012737  005324  000004          MOV     #7$,@#4         ;SET FOR TIMEOUT
1338  005316  005077  007526                  CLR     @WADRA          ;NOW TRY IT WITH WINDOW OPEN
1339  005322  000405                          BR      8$              ;SKIP THE ERROR TYPEOUT
1340  005324  017700  007474          7$:     MOV     @WCSRA,R0       ;GET FOR TYPING
1341  005330  104000                          HLT                     ;TRAPPED WITH WINDOW OPEN
1342  005332  022626                          CMP     (6)+,(6)+       ;CLEAR THE STACK
1343  005334  000406                          BR      9$              ;SKIP THE TEST
1344  005336                          8$:
1345  005336  017700  007462                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1346  005342  022700  010204                  CMP     #10204,R0       ;IS @WCSRA = 10204?
1347  005346  001401                          BEQ     .+4             ;SKIP IF OK
1348  005350  104000                          HLT                     ;@WCSRA NOT 10204
1349
1350  005352                          9$:
1351  005352  012777  000001  007444          MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1352  005360  004767  006774                  JSR     PC,TIMO         ;TIME THE RESPONCE          .. 5 ..
1353  005364  005767  007506                  TST     MM              ;IS MEMORY MANAGEMENT IN USE?
1354  005370  001403                          BEQ     15$             ;BRANCH IF NO
1355  005372  012701  020000                  MOV     #020000,R1      ;USE KIPAR1 FOR ADDR
1356  005376  000406                          BR      16$
1357  005400  016701  007574          15$:    MOV     OFFSET,R1       ;GET THE OFFSET
1358  005404  016101  015142                  MOV     ADR2(1),R1      ;SET UP THE ADDRESS
1359  005410  066701  007434                  ADD     WADRA,R1        ;ADD IN WINDOW ADDRESS
1360  005414  012737  005426  000004  16$:    MOV     #11$,@#4        ;SET FOR TIMEOUT
1361  005422  005011                          CLR     (1)             ;CAUSE A DATAO
1362  005424  000401                          BR      .+4             ;SKIP THE CLEAR
1363  005426  022626                  11$:    CMP     (6)+,(6)+       ;CLEAR THE STACK
1364  005430  017700  007370                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1365  005434  022700  150204                  CMP     #150204,R0      ;IS @WCSRA = 150204?
1366  005440  001401                          BEQ     .+4             ;SKIP IF OK
1367  005442  104000                          HLT                     ;@WCSRA NOT 150204
1368
1369  005444  016700  007354                  MOV     WCSRA,R0        ;GET THE ADDRESS
1370  005450  142760  000200  000001          BICB    #200,1(0)       ;CLEAR BIT 15
1371  005456  017700  007342                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1372  005462  022700  010204                  CMP     #10204,R0       ;IS @WCSRA = 10204?
1373  005466  001401                          BEQ     .+4             ;SKIP IF OK
1374  005470  104000                          HLT                     ;@WCSRA NOT 10204
1375
1376
1377  005472  005767  007500                  TST     MM              ;MEMORY MANAGEMENT?
1378  005476  001405                          BEQ     20$             ;BRANCH IF NO
1379  005500  016701  007474                  MOV     OFFSET,R1
1380  005504  016101  015142                  MOV     ADR2(1),R1      ;GET EXPECTED WDARA CONTENTS
1381  005510  000402                          BR      21$
1382  005512  166701  007332          20$:    SUB     WADRA,R1        ;GET RID OF THE BASE
1383  005516  017700  007310          21$:    MOV     @WDARA,R0       ;GET FOR TYPING
1384  005522  020100                          CMP     R1,R0           ;CHECK WDARA FOR ERROR ADDRESS
1385  005524  001401                          BEQ     .+4             ;SKIP IF OK
1386  005526  104000                          HLT                     ;THE WDARA CONTAINS THE WRONG ADDRESS
```

```
                                ;*****************************************************************
                                ;TEST 110                CHECK WINDOW SIZE AND WDARA
                                ;*****************************************************************
005530  104400                  TST110. SCOPE

005532  005077  007276          CLR     @WRARA          ;SET TO BANK 0
005536  012777  000001  007260  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
005544  004767  006610          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
005550  016702  007324          MOV     SIZE,R2         ;GET THE SIZE
005554  005402                  NEG     R2              ;MAKE
005556  006302                  ASL     R2              ;IT A
005560  006302                  ASL     R2              ;COUNT
005562  032767  037777  007310  BIT     #37777,SIZE     ;IS IT 32K?
005570  001002                  BNE     4$              ;SKIP IF NOT
005572  012702  120000          MOV     #120000,R2      ;FUDGE FOR 32K
005576  012737  005616  000004  4$:     MOV     #2$,@#4         ;SET FOR TIMEOUT
005604  016701  007240          MOV     WADRA,R1        ;GET THE ADDRESS OF WINDOW
005610  060102                  ADD     R1,R2           ;CREATE END OF WINDOW
005612  005011                  1$:     CLR     (1)             ;DO A DATAO
005614  000401                  BR      3$              ;SKIP ERROR
005616  022626                  2$:     CMP     (6)+,(6)+       ;CLEAR THE STACK
005620  017700  007200          3$:     MOV     @WCSRA,R0       ;GET FOR TYPING
005624  022700  110000          CMP     #110000,R0      ;ERROR FLAG SET?
005630  001401                  BEQ     .+4             ;SKIP IF OK
005632  104000                  HLT                     ;NO ACCESS ERROR - BAD WONDOW SIZE?
005634  042777  100000  007162  BIC     #100000,@WCSRA  ;CLEAR ERROR
005642  166701  007202          SUB     WADRA,R1        ;WITH RESPECT TO WINDOW
005646  017700  007160          MOV     @WDARA,R0       ;GET ADDRESS FOR TYPING
005652  020001                  CMP     R0,R1           ;CHECK ADDRESS
005654  001401                  BEQ     .+4             ;SKIP IF OK
005656  104001                  HLT     +1              ;ERROR ADDRESS IS WRONG
005660  062701  000002          ADD     #2,R1           ;GO TO NEXT
005664  066701  007160          ADD     WADRA,R1        ;MAKE IT WINDOW ADDRESS AGAIN
005670  020102                  CMP     R1,R2           ;CHECK FOR END
005672  001347                  BNE     1$              ;LOOP UNTIL DONE
005674  012737  000006  000004  MOV     #6,@#4          ;RESET 4
005702  012777  000001  007114  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
005710  004767  006444          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
```

```
1425                                    ;*****************************************************************************
1426                                    ;TEST 111                CHECK WINDOW ADDRESSING CAPABILITY
1427                                    ;*****************************************************************************
1428  005714  104400                    TST111: SCOPE
1429
1430  005716  005077  007112                    CLR     @WRARA          ;SET TO BANK 0
1431  005722  012777  000001  007074            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1432  005730  004767  006424                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1433  005734  016702  007140                    MOV     SIZE,R2         ;GET THE SIZE
1434  005740  005402                            NEG     R2              ;MAKE IT A COUNT
1435  005742  006302                            ASL     R2
1436  005744  006302                            ASL     R2
1437  005746  032767  037777  007124            BIT     #37777,SIZE     ;IS IT 32K?
1438  005754  001002                            BNE     4$              ;SKIP IF NOT
1439  005756  012702  120000                    MOV     #120000,R2      ;FUDGE FOR 32K
1440  005762  012737  006000  000004  4$:       MOV     #2$,@#4         ;SET FOR TIMEOUT
1441  005770  016701  007054                    MOV     WADRA,R1        ;GET WINDOW STARTING ADDR
1442  005774  011104                  1$:       MOV     (R1),R4         ;GO THRU THE WINDOW
1443  005776  000401                            BR      3$              ;SKIP IF ERROR
1444  006000  022626                  2$:       CMP     (6)+,(6)+       ;RESTORE STACK
1445  006002  017700  007016          3$:       MOV     @WCSRA,R0       ;GET STATUS
1446  006006  022700  150204                    CMP     #150204,R0      ;CHECK STATUS
1447  006012  001401                            BEQ     .+4             ;SKIP IF OK
1448  006014  104000                            HLT                     ;TIME OUT ERROR DID NOT OCCUR
1449  006016  042777  100000  007000            BIC     #100000,@WCSRA  ;CLEAR THE ERROR
1450  006024  012777  000001  006772            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1451  006032  004767  006322                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
1452  006036  062701  000002                    ADD     #2,R1           ;UPDATE WINDOW ADDR
1453  006042  162702  000002                    SUB     #2,R2           ;DECREMENT SIZE COUNT
1454  006046  001352                            BNE     1$              ;LOOP UNTIL DONE
1455  006050  012777  000071  006746            MOV     #71!1,@WCSRA    ;SET 71 INTO @WCSRA
1456  006056  004767  006276                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
1457                                    ;*****************************************************************************
1458                                    ;TEST 112                CHECK OPEN WINDOW READ
1459                                    ;*****************************************************************************
1460  006062  104400                    TST112: SCOPE
1461
1462  006064  012777  000001  006732            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1463  006072  004767  006262                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1464  006076  016700  006746                    MOV     WADRA,R0        ;GET WINDOW ADDRESS
1465  006102  062700  001014                    ADD     #ONEBUS,R0      ;START +ONEBUS
1466  006106  012703  001014                    MOV     #ONEBUS,R3      ;SET TO ADDRESS ONEBUS
1467  006112  016702  006762                    MOV     SIZE,R2         ;GET WINDOW SIZE
1468  006116  005402                            NEG     R2              ;MAKE IT THE LAST ADDRESS
1469  006120  006302                            ASL     R2              ;GET INTO
1470  006122  006302                            ASL     R2              ;POSITION
1471  006124  066702  006720                    ADD     WADRA,R2        ;RELOCATE TO BEGINNING OF WINDOW
1472  006130  011001                  1$:       MOV     (0),R1          ;GET THE BAD DATA
1473  006132  021013                            CMP     (0),(3)         ;CHECK THE DATA
1474  006134  001401                            BEQ     .+4             ;SKIP IF SAME
1475  006136  104001                            HLT     +1              ;WINDOW DATA NOT BANK 0 DATA
1476  006140  022320                            CMP     (3)+,(0)+       ;ADD 2 TO BOTH
1477  006142  022703  010330                    CMP     #DONE,R3        ;END OF DATA?
1478  006146  001402                            BEQ     2$              ;YES - GET OUT
1479  006150  020200                            CMP     R2,R0           ;END OF WINDOW?
1480  006152  001366                            BNE     1$              ;NO - LOOP
```

```
1481  006154                                  2$:
1482  006154  012777  000001  006642                MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1483  006162  004767  006172                        JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
1484
1485
1486                                  ;*****************************************************************************
1487                                  ;TEST 113                   CHECK OPEN WINDOW WRITE
1488                                  ;*****************************************************************************
1489  006166  104400                  TST113: SCOPE
1490
1491  006170  012777  000001  006626          MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1492  006176  004767  006156                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1493  006202  016700  006642                  MOV     WADRA,R0        ;GET  WINDOW ADDRESS
1494  006206  062700  000500                  ADD     #500,R0         ;START AT 500
1495  006212  012703  000100                  MOV     #100,R3         ;SET COUNT (500-676)
1496  006216  010320                  1$:     MOV     R3,(0)+         ;LOAD THE DATA
1497  006220  005303                          DEC     R3              ;DEC THE COUNT
1498  006222  001375                          BNE     1$              ;LOOP UNTIL DONE
1499  006224  162700  000200                  SUB     #200,R0         ;BACK IT UP
1500  006230  012703  000100                  MOV     #100,R3         ;SET COUNT
1501  006234  011001                  2$:     MOV     (0),R1          ;GET DATA FOR TYPING
1502  006236  021003                          CMP     (0),R3          ;CHECK DATA
1503  006240  001401                          BEQ     .+4             ;SKIP IF OK
1504  006242  104001                          HLT     +1              ;DATA IS WRONG
1505  006244  005720                          TST     (0)+            ;ADD 2
1506  006246  005303                          DEC     R3              ;DEC THE COUNT
1507  006250  001371                          BNE     2$              ;LOOP UNTIL END
```

```
1508                                    ;*****************************************************************
1509                                    ;TEST 114                   CHECK INSTRUCTION EXECUTION
1510                                    ;*****************************************************************
1511  006252  104400                    TST114: SCOPE
1512
1513  006254  012777  000001  006542            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1514  006262  004767  006072                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1515  006266  016700  006556                    MOV     WADRA,R0        ;GET WINDOW ADDRESS
1516  006272  004760  000700                    JSR     PC,CODE1(0)     ;EXECUTE CODE IN OTHER MACHINE
1517
1518                                    ;*****************************************************************
1519                                    ;TEST 115                   CHECK INTERUPTS
1520                                    ;*****************************************************************
1521  006276  104400                    TST115: SCOPE
1522
1523  006300  012777  006332  006536            MOV     #4$,@WVECA      ;SET FOR TRAP
1524  006306  013777  177776  006532            MOV     @#PS,@WVECA+2   ;SET SAME PS
1525  006314  012777  000101  006502            MOV     #100!1,@WCSRA   ;SET 100 INTO @WCSRA
1526  006322  004767  006032                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1527  006326  104000                            HLT                     ;B DID NOT INTERUPT A
1528  006330  000401                            BR      .+4             ;SKIP ON ERROR
1529  006332  022626                    4$:     CMP     (6)+,(6)+       ;CLEAR STACK
1530  006334  005726                            TST     (6)+            ;CLEAR THE REST
1531  006336  017700  006462                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1532  006342  022700  010100                    CMP     #10100,R0       ;IS @WCSRA = 10100?
1533  006346  001401                            BEQ     .+4             ;SKIP IF OK
1534  006350  104000                            HLT                     ;@WCSRA NOT 10100
1535  006352  016777  006470  006464            MOV     WVECA+2,@WVECA  ;RESET FOR
1536  006360  005077  006462                    CLR     @WVECA+2        ;ILL TRAP
1537
1538
1539  006364  012777  000001  006432            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1540  006372  004767  005762                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
1541
1542  006376  012701  000045                    MOV     #45,R1          ;SET FOR DOUBLE LOOP
1543  006402  005000                    20$:    CLR     R0              ;WAIT
1544  006404  005300                    10$:    DEC     R0              ;BEFORE
1545  006406  001376                            BNE     10$             ;RESTARTING
1546  006410  005301                            DEC     R1              ;DEC COUNT
1547  006412  001373                            BNE     20$             ;LOOP UNTIL DONE
1548
1549  006414  000167  001710                    JMP     DONE            ;LOOP
```

```
1550  006420  112767  000200  172352  SECOND: MOVB    #200,ICNT       ;SET TEST 201
1551
1552                                  ;*************************************************************************
1553                                  ;TEST 201               CHECK R/W OF WCSRA
1554                                  ;*************************************************************************
1555  006426  104400                  TST201: SCOPE
1556
1557  006430  012777  010000  006365          MOV     #10000,@WCSRA   ;LOAD 10000 INTO @WCSRA
1558  006436  017700  006362                  MOV     @WCSRA,R0       ;GET FOR TYPING
1559  006442  022700  010000                  CMP     #10000,R0       ;CHECK @WCSRA FOR 10000
1560  006446  001401                          BEQ     .+4             ;SKIP IF OK
1561  006450  104000                          HLT                     ;@WCSRA DOES NOT CONTAIN 10000
1562
1563  006452  012777  010572  006344          MOV     #10572,@WCSRA   ;LOAD 10572 INTO @WCSRA
1564  006460  017700  006340                  MOV     @WCSRA,R0       ;GET FOR TYPING
1565  006464  022700  010572                  CMP     #10572,R0       ;CHECK @WCSRA FOR 10572
1566  006470  001401                          BEQ     .+4             ;SKIP IF OK
1567  006472  104000                          HLT                     ;@WCSRA DOES NOT CONTAIN 10572
1568
1569  006474  012777  010452  006322          MOV     #10452,@WCSRA   ;LOAD 10452 INTO @WCSRA
1570  006502  017700  006316                  MOV     @WCSRA,R0       ;GET FOR TYPING
1571  006506  022700  010452                  CMP     #10452,R0       ;CHECK @WCSRA FOR 10452
1572  006512  001401                          BEQ     .+4             ;SKIP IF OK
1573  006514  104000                          HLT                     ;@WCSRA DOES NOT CONTAIN 10452
1574
1575  006516  012777  010120  006300          MOV     #10120,@WCSRA   ;LOAD 10120 INTO @WCSRA
1576  006524  017700  006274                  MOV     @WCSRA,R0       ;GET FOR TYPING
1577  006530  022700  010120                  CMP     #10120,R0       ;CHECK @WCSRA FOR 10120
1578  006534  001401                          BEQ     .+4             ;SKIP IF OK
1579  006536  104000                          HLT                     ;@WCSRA DOES NOT CONTAIN 10120
1580  006540  012777  010000  006256          MOV     #10000,@WCSRA   ;ZERO THE REGISTER
```

```
                                        ;*****************************************************************************
                                        ;TEST 202                   CHECK R/W OF WDBRA
                                        ;*****************************************************************************
006546  104400                          TST202: SCOPE

006550  012777  177777  006250          MOV     #-1,@WDBRA      ;SET THE SETABLE BITS
006556  017700  006244                  MOV     @WDBRA,R0       ;GET THE RESULT
006562  022700  177777                  CMP     #-1!0,R0        ;CHECK FOR PROPER BITS
006566  001401                          BEQ     .+4             ;SKIP IF OK
006570  104000                          HLT                     ;WDBRA NOT -1!0

006572  012777  052525  006226          MOV     #52525,@WDBRA   ;SET THE SETABLE BITS
006600  017700  006222                  MOV     @WDBRA,R0       ;GET THE RESULT
006604  022700  052525                  CMP     #52525!0,R0     ;CHECK FOR PROPER BITS
006610  001401                          BEQ     .+4             ;SKIP IF OK
006612  104000                          HLT                     ;WDBRA NOT 52525!0

006614  012777  125252  006204          MOV     #125252,@WDBRA  ;SET THE SETABLE BITS
006622  017700  006200                  MOV     @WDBRA,R0       ;GET THE RESULT
006626  022700  125252                  CMP     #125252!0,R0    ;CHECK FOR PROPER BITS
006632  001401                          BEQ     .+4             ;SKIP IF OK
006634  104000                          HLT                     ;WDBRA NOT 125252!0

006636  005077  006164                  CLR     @WDBRA          ;CLEAR ALL SETABLE BITS
006642  017700  006160                  MOV     @WDBRA,R0       ;GET THE RESULT
006646  022700  000000                  CMP     #0,R0           ;CHECK FOR PROPER BITS
006652  001401                          BEQ     .+4             ;SKIP IF OK
006654  104000                          HLT                     ;WDBRA NOT 0
```

```
                                ;*****************************************************************
                                ;TEST 203               CHECK R/W OF WRARA
                                ;*****************************************************************
006656  104400                  TST203: SCOPE

006660  016702  006214                MOV     SIZE,R2         ;GET WINDOW SIZE
006664  005102                        COM     R2              ;MAKE IT A MASK

006666  012703  000000                MOV     #0,R3           ;GET THE DATA
006672  040203                        BIC     R2,R3           ;CLEAR THE UNUSED BITS
006674  010377  006134                MOV     R3,@WRARA       ;SET THE BITS IN WRARA
006700  017700  006130                MOV     @WRARA,R0       ;GET THE DATA
006704  040200                        BIC     R2,R0           ;CLEAR THE JUNK
006706  020300                        CMP     R3,R0           ;CHECK THE DATA
006710  001401                        BEQ     .+4             ;SKIP IF OK
006712  104000                        HLT                     ;WRARA NOT 0

006714  012703  177777                MOV     #-1,R3          ;GET THE DATA
006720  040203                        BIC     R2,R3           ;CLEAR THE UNUSED BITS
006722  010377  006106                MOV     R3,@WRARA       ;SET THE BITS IN WRARA
006726  017700  006102                MOV     @WRARA,R0       ;GET THE DATA
006732  040200                        BIC     R2,R0           ;CLEAR THE JUNK
006734  020300                        CMP     R3,R0           ;CHECK THE DATA
006736  001401                        BEQ     .+4             ;SKIP IF OK
006740  104000                        HLT                     ;WRARA NOT -1

006742  012703  052525                MOV     #52525,R3       ;GET THE DATA
006746  040203                        BIC     R2,R3           ;CLEAR THE UNUSED BITS
006750  010377  006060                MOV     R3,@WRARA       ;SET THE BITS IN WRARA
006754  017700  006054                MOV     @WRARA,R0       ;GET THE DATA
006760  040200                        BIC     R2,R0           ;CLEAR THE JUNK
006762  020300                        CMP     R3,R0           ;CHECK THE DATA
006764  001401                        BEQ     .+4             ;SKIP IF OK
006766  104000                        HLT                     ;WRARA NOT 52525

006770  012703  125252                MOV     #125252,R3      ;GET THE DATA
006774  040203                        BIC     R2,R3           ;CLEAR THE UNUSED BITS
006776  010377  006032                MOV     R3,@WRARA       ;SET THE BITS IN WRARA
007002  017700  006026                MOV     @WRARA,R0       ;GET THE DATA
007006  040200                        BIC     R2,R0           ;CLEAR THE JUNK
007010  020300                        CMP     R3,R0           ;CHECK THE DATA
007012  001401                        BEQ     .+4             ;SKIP IF OK
007014  104000                        HLT                     ;WRARA NOT 125252
007016  005077  006012                CLR     @WRARA          ;CLEAR JUNK
```

```
1653                                    ;*************************************************************************
1654                                    ;TEST 204                   CHECK CROSS COMMUNICATION BITS
1655                                    ;*************************************************************************
1656  007022  104400                    TST204: SCOPE
1657
1658  007024  005037  000006                    CLR     @#6             ;CLEAR THE STATUS WORD
1659  007030  005037  177776                    CLR     @#PS            ;SPL 0
1660  007034  012777  000001  005762            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1661  007042  004767  005312                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1662
1663  007046  017700  005752                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1664  007052  022700  010000                    CMP     #10000,R0       ;IS @WCSRA = 10000?
1665  007056  001401                            BEQ     .+4             ;SKIP IF OK
1666  007060  104000                            HLT                     ;@WCSRA NOT 10000
1667  007062  012777  000001  005734            MOV     #0!1,@WCSRA     ;SET 0 INTO @WCSRA
1668  007070  004767  005264                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
1669
1670  007074  017700  005724                    MOV     @WCSRA,R0        GET @WCSRA FOR TYPING
1671  007100  022700  011000                    CMP     #11000,R0       ;IS @WCSRA = 11000?
1672  007104  001401                            BEQ     .+4             ;SKIP IF OK
1673  007106  104000                            HLT                     ;@WCSRA NOT 11000
1674  007110  012777  000011  005706            MOV     #10!1,@WCSRA    ;SET 10 INTO @WCSRA
1675  007116  004767  005236                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
1676
1677  007122  017700  005676                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1678  007126  022700  012010                    CMP     #12010,R0       ;IS @WCSRA = 12010?
1679  007132  001401                            BEQ     .+4             ;SKIP IF OK
1680  007134  104000                            HLT                     ;@WCSRA NOT 12010
1681  007136  012777  000021  005660            MOV     #20!1,@WCSRA    ;SET 20 INTO @WCSRA
1682  007144  004767  005210                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 4 ..
1683
1684  007150  017700  005650                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1685  007154  022700  013020                    CMP     #13020,R0       ;IS @WCSRA = 13020?
1686  007160  001401                            BEQ     .+4             ;SKIP IF OK
1687  007162  104000                            HLT                     ;@WCSRA NOT 13020
1688  007164  012777  000031  005632            MOV     #30!1,@WCSRA    ;SET 30 INTO @WCSRA
1689  007172  004767  005162                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 5 ..
1690
1691  007176  017700  005622                    MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1692  007202  022700  014030                    CMP     #14030,R0       ;IS @WCSRA = 14030?
1693  007206  001401                            BEQ     .+4             ;SKIP IF OK
1694  007210  104000                            HLT                     ;@WCSRA NOT 14030
1695  007212  012777  000041  005604            MOV     #40!1,@WCSRA    ;SET 40 INTO @WCSRA
1696  007220  004767  005134                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 6 ..
1697
```

```
1698  007224  017700  005574          MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1699  007230  022700  015040          CMP     #15040,R0       ;IS @WCSRA = 15040?
1700  007234  001401                  BEQ     .+4             ;SKIP IF OK
1701  007236  104000                  HLT                     ;@WCSRA NOT 15040
1702  007240  012777  000051  005556  MOV     #50!1,@WCSRA    ;SET 50 INTO @WCSRA
1703  007246  004767  005106          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 7 ..
1704
1705  007252  017700  005546          MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1706  007256  022700  016050          CMP     #16050,R0       ;IS @WCSRA = 16050?
1707  007262  001401                  BEQ     .+4             ;SKIP IF OK
1708  007264  104000                  HLT                     ;@WCSRA NOT 16050
1709  007266  012777  000061  005530  MOV     #60!1,@WCSRA    ;SET 60 INTO @WCSRA
1710  007274  004767  005060          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 10 ..
1711
1712  007300  017700  005520          MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1713  007304  022700  017060          CMP     #17060,R0       ;IS @WCSRA = 17060?
1714  007310  001401                  BEQ     .+4             ;SKIP IF OK
1715  007312  104000                  HLT                     ;@WCSRA NOT 17060
1716  007314  012777  000071  005502  MOV     #70!1,@WCSRA    ;SET 70 INTO @WCSRA
1717  007322  004767  005032          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 11 ..
1718
1719  007326  017700  005472          MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1720  007332  022700  010074          CMP     #10074,R0       ;IS @WCSRA = 10074?
1721  007336  001401                  BEQ     .+4             ;SKIP IF OK
1722  007340  104000                  HLT                     ;@WCSRA NOT 10074
1723  007342  012777  000003  005454  MOV     #2!1,@WCSRA     ;SET 2 INTO @WCSRA
1724  007350  004767  005004          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 12 ..
1725
1726  007354  017700  005444          MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
1727  007360  022700  010202          CMP     #10202,R0       ;IS @WCSRA = 10202?
1728  007364  001401                  BEQ     .+4             ;SKIP IF OK
1729  007366  104000                  HLT                     ;@WCSRA NOT 10202
1730  007370  012777  000401  005426  MOV     #400!1,@WCSRA   ;SET 400 INTO @WCSRA
```

```
                                        ;*****************************************************************************
                                        ;TEST 205                 CHECK CROSS WDBR'S WITH WINDOW CLOSED
                                        ;*****************************************************************************
007376  104400                          TST205: SCOPE

007400  004767  004754                          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
007404  012777  000001  005412                  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
007412  004767  004742                          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
007416  017700  005406                          MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
007422  022700  177777                          CMP     #-1,R0          ;IS @WDBRB = -1?
007426  001401                                  BEQ     .+4             ;SKIP IF OK
007430  104000                                  HLT                     ;@WDBRB NOT -1

007432  012777  177777  005366                  MOV     #-1,@WDBRA      ;LOAD -1 INTO @WDBRA
007440  012777  000001  005356                  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
007446  004767  004706                          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
007452  017700  005352                          MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
007456  022700  000000                          CMP     #0,R0           ;IS @WDBRB = 0?
007462  001401                                  BEQ     .+4             ;SKIP IF OK
007464  104000                                  HLT                     ;@WDBRB NOT 0

007466  012777  000000  005332                  MOV     #0,@WDBRA       ;LOAD 0 INTO @WDBRA
007474  012777  000001  005322                  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
007502  004767  004652                          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 4 ..
007506  017700  005316                          MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
007512  022700  052525                          CMP     #52525,R0       ;IS @WDBRB = 52525?
007516  001401                                  BEQ     .+4             ;SKIP IF OK
007520  104000                                  HLT                     ;@WDBRB NOT 52525

007522  012777  052525  005276                  MOV     #52525,@WDBRA   ;LOAD 52525 INTO @WDBRA
007530  012777  000001  005266                  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
007536  004767  004616                          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 5 ..
007542  017700  005262                          MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
007546  022700  125252                          CMP     #125252,R0      ;IS @WDBRB = 125252?
007552  001401                                  BEQ     .+4             ;SKIP IF OK
007554  104000                                  HLT                     ;@WDBRB NOT 125252

007556  012777  125252  005242                  MOV     #125252,@WDBRA  ;LOAD 125252 INTO WDBRA
007564  012777  000001  005232                  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
```

```
                                ;*****************************************************************
                                ;TEST 206                CHECK CROSS WDBR'S WITH WINDOW OPEN AND CLOSED
                                ;*****************************************************************
007572  104400                  TST206: SCOPE

007574  004767  004560          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
007600  012777  000401  005216  MOV     #400!1,@WCSRA   ;SET 400 INTO @WCSRA
007606  004767  004546          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
007612  017700  005212          MOV     @WDBRB,R0       ;GET @WDBRB FOR TYPING
007616  022700  125252          CMP     #125252,R0      ;IS @WDBRB = 125252?
007622  001401                  BEQ     .+4             ;SKIP IF OK
007624  104000                  HLT                     ;@WDBRB NOT 125252

007626  012777  052525  005172  MOV     #52525,@WDBRA   ;LOAD 52525 INTO @WDBRA
007634  017700  005166          MOV     @WDBRA,R0       ;GET @WDBRA FOR TYPING
007640  022700  125252          CMP     #125252,R0      ;IS @WDBRA = 125252?
007644  001401                  BEQ     .+4             ;SKIP IF OK
007646  104000                  HLT                     ;@WDBRA NOT 125252

007650  012777  000001  005146  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
007656  004767  004476          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
007662  012777  052525  005136  MOV     #52525,@WDBRA   ;LOAD 52525 INTO @WDBRA
007670  012777  000001  005126  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
007676  004767  004456          JSR     PC,TIMO         ;TIME THE RESPONCE      .. 4 ..
007702  017700  005120          MOV     @WDBRA,R0       ;GET @WDBRA FOR TYPING
007706  022700  052525          CMP     #52525,R0       ;IS @WDBRA = 52525?
007712  001401                  BEQ     .+4             ;SKIP IF OK
007714  104000                  HLT                     ;@WDBRA NOT 52525
007716  012777  000001  005100  MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
```

```
                                ;*****************************************************************************
                                ;TEST 207                CHECK ERROR CONDITIONS AND LOCKOUT
                                ;*****************************************************************************
007724  104400                  TST207: SCOPE

007726  004767  004426                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
007732  005077  005076                  CLR     @WRARA          ;MAP TO OK BANK
007736  012777  000001  005060          MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
007744  004767  004410                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
007750  012777  000401  005046          MOV     #400!1,@WCSRA   ;SET 400 INTO @WCSRA
007756  004767  004376                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 3 ..
007762  012777  000003  005034          MOV     #2!1,@WCSRA     ;SET 2 INTO @WCSRA
007770  004767  004364                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 4 ..
007774  012777  000403  005022          MOV     #402!1,@WCSRA   ;SET 402 INTO @WCSRA
010002  004767  004352                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 5 ..
010006  016705  005166                  MOV     OFFSET,TTY      ;GET OFFSET
010012  016577  015124  005014          MOV     ADR1(TTY),@WRARA;SET TO PERF PAGE
010020  012777  000403  004776          MOV     #402!1,@WCSRA   ;SET 402 INTO @WCSRA


                                ;*****************************************************************************
                                ;TEST 210                CHECK WINDOW SIZE AND WDARA
                                ;*****************************************************************************
010026  104400                  TST210: SCOPE

010030  004767  004324                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
010034  005077  004774                  CLR     @WRARA          ;MAP TO 0
010040  012777  000001  004756          MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
010046  032777  010000  004750          BIT     #10000,@WCSRA   ;NEW DATA SET?
010054  001774                          BEQ     .-6             ;HANG
010056  012777  000001  004740          MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA


                                ;*****************************************************************************
                                ;TEST 211                CHECK WINDOW ADDRESSING CAPABILITY
                                ;*****************************************************************************
010064  104400                  TST211: SCOPE

010066  004767  004266                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
010072  005067  005004                  CLR     DISP            ;INITIALIZE EXP DISPLACEMENT ADDR
010076  017777  004734  004730          MOV     @WADRAX,@WRARA  ;FORCE THE WINDOW TO RELOCATE TO ITSELF
010104                          1$:
010104  012777  000403  004712          MOV     #402!1,@WCSRA   ;SET 402 INTO @WCSRA
010112  004767  004242                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
010116  032777  007000  004700          BIT     #7000,@WCSRA    ;IS TEST COMPLETE?
010124  001024                          BNE     2$              ;BRANCH IF YES
010126  017700  004672                  MOV     @WCSRA,R0       ;GET STATUS
010132  022700  110402                  CMP     #110402,R0      ;IS IT CORRECT?
010136  001401                          BEQ     .+4             ;SKIP IF OK
010140  104000                          HLT                     ;NO TIMEOUT ERROR
010142  042777  100000  004654          BIC     #100000,@WCSRA  ;CLEAR THE ERROR
010150  017700  004656                  MOV     @WDARA,R0       ;GET DISPLACEMENT ADDR
010154  016701  004722                  MOV     DISP,R1         ;GET EXP DISPLACEMENT ADDR
010160  020001                          CMP     R0,R1           ;IS DISPLACEMENT ADDR CORRECT?
010162  001401                          BEQ     .+4             ;SKIP IF OK
010164  104001                          HLT     +1              ;DISPLACEMENT ADDR INCORRECT
```

```
1855  010166  062767  000002  004706          ADD     #2,DISP         ;UPDATE EXPECTED ADDR
1856  010174  000743                          BR      1$              ;LOOP
1857  010176                          2$:
1858  010176  012777  000001  004620          MOV     #!!1,@WCSRA     ;SET 1 INTO @WCSRA
1859
1860
1861                                  ;*****************************************************************************
1862                                  ;TEST 212                  CHECK OPEN WINDOW READ
1863                                  ;*****************************************************************************
1864  010204  104400                  TST212: SCOPE
1865
1866  010206  004767  004146                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1867  010212  005077  004616                  CLR     @WRARA          ;SET TO BANK 0
1868  010216  012777  000403  004600          MOV     #402!1,@WCSRA   ;SET 402 INTO @WCSRA
1869  010224  032777  010000  004572          BIT     #10000,@WCSRA   ;NEW DATA SET?
1870  010232  001774                          BEQ     .-6             ;HANG
1871  010234  012777  000001  004562          MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1872
1873
1874                                  ;*****************************************************************************
1875                                  ;TEST 213                  CHECK OPEN WINDOW  WRITE
1876                                  ;*****************************************************************************
1877  010242  104400                  TST213: SCOPE
1878
1879  010244  004767  004110                  JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1880  010250  005077  004560                  CLR     @WRARA          ;SET TO BANK 0
1881  010254  012777  000403  004542          MOV     #402!1,@WCSRA   ;SET 402 INTO @WCSRA
```

```
1882                                    ;*************************************************************************
1883                                    ;TEST 214               CHECK INSTRUCTION EXECUTION
1884                                    ;*************************************************************************
1885  010262  104400                    TST214: SCOPE
1886
1887  010264  004767  004070                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1888
1889  010270  005077  004540                    CLR     @WRARA          ;SET TO BANK 0
1890  010274  012777  000403  004522            MOV     #402!1,@WCSRA   ;SET 402 INTO @WCSRA
1891
1892                                    ;*************************************************************************
1893                                    ;TEST 215               CHECK INTERUPTS
1894                                    ;*************************************************************************
1895  010302  104400                    TST215: SCOPE
1896
1897  010304  004767  004050                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 1 ..
1898  010310  012777  000001  004506            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
1899
1900  010316  004767  004036                    JSR     PC,TIMO         ;TIME THE RESPONCE      .. 2 ..
1901  010322  012777  000001  004474            MOV     #1!1,@WCSRA     ;SET 1 INTO @WCSRA
```

```
 1902  010330  104400                    DONE:   SCOPE
 1903  010332  062767  000001  170446            ADD     #1,PCNT+2       ;ADD ONE TO THE PASS COUNT
 1904  010340  005567  170440                    ADC     PCNT            ;MAKE IT DOUBLE PREC.
 1905  010344  032737  002000  177570            BIT     #SW10,@#SWR     ;RING THE BELL?
 1906  010352  001002                            BNE     4$              ;NO
 1907  010354  000004  000007                    TYPE    ,BELL           ;RING THE BELL
 1908  010360  013700  000042            4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
 1909  010364  001404                            BEQ     3$              ;BRANCH IF NONE
 1910  010366  004710                            JSR     7,(0)           ;GO TO MONITOR
 1911  010370  000240                            NOP
 1912  010372  000240                            NOP
 1913  010374  000240                            NOP
 1914  010376  000167  173020            3$:     JMP     GO              ;RETURN
 1915
 1916
 1917
 1918
 1919                                    ;       $TYPE           MESSAGE TYPEOUT ROUTINE
 1920
 1921                                    ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
 1922                                    ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
 1923                                    ;MESSAGE STARTING IN LOCATION "ADR:", 2) "TYPE ,CHAR" - TYPES
 1924                                    ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
 1925                                    ;THE MESSAGE WHICH IS INLINE ASCII.
 1926
 1927  010402  010467  000146            IOT$:   MOV     R4,IOT1$        ;SAVE R4
 1928  010406  010546                            MOV     TTY,-(6)        ;SAVE TTY R5=TTY
 1929  010410  017605  000002                    MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
 1930  010414  032705  177400                    BIT     #177400,TTY     ;IS IT A TYPEM?
 1931  010420  001004                            BNE     1$              ;NO
 1932  010422  010567  000124                    MOV     TTY,.TYPE       ;GET THE CHARACTER
 1933  010426  012705  010552                    MOV     #.TYPE,TTY      ;FUDGE THE ADDRESS
 1934  010432  105715                    1$:     TSTB    (TTY)           ;TERMINATOR?
 1935  010434  001423                            BEQ     2$              ;GET OUT IF SO
 1936  010436  122715  000012                    CMPB    #12,(TTY)       ;IS THE CHAR A LINE FEED?
 1937  010442  001012                            BNE     4$              ;BRANCH IF NO
 1938  010444  116704  000101                    MOVB    FILCHR+1,R4     ;GET FILLER COUNT
 1939  010450  116737  000074  177566    5$:     MOVB    FILCHR,@#177566 ;OUTPUT FILLER CHAR
 1940  010456  105737  177564                    TSTB    @#177564        ;WAIT FOR DONE
 1941  010462  100375                            BPL     .-4
 1942  010464  005304                            DEC     R4              ;DECREMENT FILLER COUNT
 1943  010466  001370                            BNE     5$
 1944  010470  112537  177566            4$:     MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
 1945  010474  105737  177564                    TSTB    @#177564        ;IS THE PRINTER READY
 1946  010500  100375                            BPL     .-4             ;WAIT UNTIL IT IS
 1947  010502  000753                            BR      1$              ;GET THE NEXT CHARACTER
 1948  010504  017646  000002            2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
 1949  010510  062766  000002  000004            ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
 1950  010516  022666  000002                    CMP     (6)+,2(6)       ;IS IT .+2?
 1951  010522  001006                            BNE     3$              ;NO
 1952  010524  062705  000002                    ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
 1953  010530  042705  000001                    BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
 1954  010534  010566  000002                    MOV     TTY,2(6)        ;RESTORE ADDRESS
 1955  010540  012605                    3$:     MOV     (6)+,TTY        ;RESTORE TTY
 1956  010542  016704  000006                    MOV     IOT1$,R4        ;RESTORE R4
 1957  010546  000002                            RTI                     ;RETURN
```

```
  1958  010550  001000                    FILCHR: 1U00                    ;FILCHR=0 (CHAR)  FILCHR+1=2 (COUNT)
  1959  010552  000000                    .TYPE:  0                       ;CHARACTER TYPE LOCATION
  1960  010554  000000                    IOT1$:  0
```

```
 1961  010556  012767  000300  170214  GO.HST: MOV     #300,ICNT       ;HOST COMPUTER ENTRY
 1962  010564  012706  000500                  MOV     #500,SP         ;INITIALIZE STACK
 1963
 1964                                  ;*****************************************************************
 1965                                  ;TEST 301                 POWER FAIL TEST - PF - GOES DOWN - BIT13 SET
 1966                                  ;*****************************************************************
 1967  010570  104400                  TST301: SCOPE
 1968
 1969  010572  012737  010630  000024          MOV     #1$,@#24        ;SET FOR POWER FAIL
 1970  010600  012737  000340  000026          MOV     #340,@#26       ;LOCK IT UP
 1971  010606  012777  010672  004230          MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
 1972  010614  012777  000502  004202          MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
 1973  010622  000001                          WAIT                    ;WAIT FOR INTERUPT
 1974  010624  000000                          HALT                    ;RETURN FROM TRAP
 1975  010626  000776                          BR      .-2             ;LOCK IT UP
 1976  010630  022626                  1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
 1977  010632  012777  010652  004352          MOV     #3$,@PUVEC$     ;RESET UP VECTOR
 1978  010640  017746  004160                  MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
 1979  010644  010667  004340                  MOV     SP,SAVE6        ;SAVE THE SP
 1980  010650  000000                          HALT                    ;WAIT FOR POWER FAIL
 1981  010652  016706  004332          3$:     MOV     SAVE6,SP        ;RESTORE THE SP
 1982  010656  012600                          MOV     (6)+,R0         ;GET FOR TYPING
 1983  010660  022700  120706                  CMP     #120706,R0      ;CHECK THE WCSRA
 1984  010664  001401                          BEQ     .+4             ;SKIP IF OK
 1985  010666  104000                          HLT                     ;WCSRA NOT 120706 AFTER PF
 1986  010670  000404                          BR      4$              ;SKIP WINDOW AREA
 1987  010672  022626                  2$:     CMP     (6)+,(6)+       ;CLEAR STACK
 1988  010674  017700  004124                  MOV     @WCSRA,R0       ;GET FOR TYPING
 1989  010700  104000                          HLT                     ;SHOULD NOT HAVE TRAPPED
 1990  010702  012737  000026  000024  4$:     MOV     #26,@#24        ;RESET PF VECTOR
 1991  010710  005037  177776                  CLR     @#PS            ;LOWER PROCESSOR STATUS
```

```
                                ;*****************************************************************************
                                ;TEST 302                POWER FAIL TEST - NO PF - STAYS UP - BIT13 SET
                                ;*****************************************************************************
010714  104400                  TST302: SCOPE

010716  012737  010754  000024          MOV     #1$,@#24        ;SET FOR POWER FAIL
010724  012737  000340  000026          MOV     #340,@#26       ;LOCK IT UP
010732  012777  011010  004104          MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
010740  012777  000502  004056          MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
010746  000001                        WAIT                    ;WAIT FOR INTERUPT
010750  000000                        HALT                    ;RETURN FROM TRAP
010752  000776                        BR      .-2             ;LOCK IT UP
010754  022626                  1$:   CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
010756  012777  010776  004226        MOV     #3$,@PUVEC$     ;RESET UP VECTOR
010764  017746  004034                MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
010770  010667  004214                MOV     SP,SAVE6        ;SAVE THE SP
010774  000000                        HALT                    ;WAIT FOR POWER FAIL
010776  016706  004206          3$:   MOV     SAVE6,SP        ;RESTORE THE SP
011002  012600                        MOV     (6)+,R0         ;GET FOR TYPING
011004  104000                        HLT                     ;DID NOT POWER FAIL
011006  000407                        BR      4$              ;SKIP WINDOW AREA
011010  022626                  2$:   CMP     (6)+,(6)+       ;CLEAR STACK
011012  017700  004006                MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
011016  022700  120706                CMP     #120706,R0      ;IS @WCSRA = 120706?
011022  001401                        BEQ     .+4             ;SKIP IF OK
011024  104000                        HLT                     ;@WCSRA NOT 120706
011026  012737  000026  000024  4$:   MOV     #26,@#24        ;RESET PF VECTOR
011034  000004  011040                TYPE    ,..+2           ;.ASCIZ <15><12>"END OF TEST"
011056  000000                        HALT
```

```
 2021  011060  012767  000310  167712  GO.OTH: MOV     #310,ICNT       ;OTHER COMPUTER ENTRY
 2022
 2023                                  ;*************************************************************************
 2024                                  ;TEST 311                 POWER FAIL TEST - PF - GOES DOWN - BIT13 SET
 2025                                  ;*************************************************************************
 2026  011066  104400                  TST311: SCOPE
 2027
 2028  011070  012737  011126  000024          MOV     #1$,@#24        ;SET FOR POWER FAIL
 2029  011076  012737  000340  000026          MOV     #340,@#26       ;LOCK IT UP
 2030  011104  012777  011170  003732          MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
 2031  011112  012777  000502  003704          MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
 2032  011120  000001                          WAIT                    ;WAIT FOR INTERUPT
 2033  011122  000000                          HALT                    ;RETURN FROM TRAP
 2034  011124  000776                          BR      .-2             ;LOCK IT UP
 2035  011126  022626                  1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
 2036  011130  012777  011150  004054          MOV     #3$,@PUVEC$     ;RESET UP VECTOR
 2037  011136  017746  003662                  MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
 2038  011142  010667  004042                  MOV     SP,SAVE6        ;SAVE THE SP
 2039  011146  000000                          HALT                    ;WAIT FOR POWER FAIL
 2040  011150  016706  004034          3$:     MOV     SAVE6,SP        ;RESTORE THE SP
 2041  011154  012600                          MOV     (6)+,R0         ;GET FOR TYPING
 2042  011156  022700  120706                  CMP     #120706,R0      ;CHECK THE WCSRA
 2043  011162  001401                          BEQ     .+4             ;SKIP IF OK
 2044  011164  104000                          HLT                     ;WCSRA NOT 120706 AFTER PF
 2045  011166  000404                          BR      4$              ;SKIP WINDOW AREA
 2046  011170  022626                  2$:     CMP     (6)+,(6)+       ;CLEAR STACK
 2047  011172  017700  003626                  MOV     @WCSRA,R0       ;GET FOR TYPING
 2048  011176  104000                          HLT                     ;SHOULD NOT HAVE TRAPPED
 2049  011200  012737  000026  000024  4$:     MOV     #26,@#24        ;RESET PF VECTOR
```

```
2050                                    ;*****************************************************************************
2051                                    ;TEST 312               POWER FAIL TEST - PF - GOES DOWN - BIT13 NOT SET
2052                                    ;*****************************************************************************
2053  011206  104400                    TST312: SCOPE
2054
2055  011210  012737  011246  000024            MOV     #1$,@#24        ;SET FOR POWER FAIL
2056  011216  012737  000340  000026            MOV     #340,@#26       ;LOCK IT UP
2057  011224  012777  011310  003612            MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
2058  011232  012777  000502  003564            MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
2059  011240  000001                            WAIT                    ;WAIT FOR INTERUPT
2060  011242  000000                            HALT                    ;RETURN FROM TRAP
2061  011244  000776                            BR      .-2             ;LOCK IT UP
2062  011246  022626                    1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
2063  011250  012777  011270  003734            MOV     #3$,@PUVEC$     ;RESET UP VECTOR
2064  011256  017746  003542                    MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
2065  011262  010667  003722                    MOV     SP,SAVE6        ;SAVE THE SP
2066  011266  000000                            HALT                    ;WAIT FOR POWER FAIL
2067  011270  016706  003714            3$:     MOV     SAVE6,SP        ;RESTORE THE SP
2068  011274  012600                            MOV     (6)+,R0         ;GET FOR TYPING
2069  011276  022700  000706                    CMP     #706,R0         ;CHECK THE WCSRA
2070  011302  001401                            BEQ     .+4             ;SKIP IF OK
2071  011304  104000                            HLT                     ;WCSRA NOT 706 AFTER PF
2072  011306  000404                            BR      4$              ;SKIP WINDOW AREA
2073  011310  022626                    2$:     CMP     (6)+,(6)+       ;CLEAR STACK
2074  011312  017700  003506                    MOV     @WCSRA,R0       ;GET FOR TYPING
2075  011316  104000                            HLT                     ;SHOULD NOT HAVE TRAPPED
2076  011320  012737  000026  000024    4$:     MOV     #26,@#24        ;RESET PF VECTOR
2077  011326  000004  011332                    TYPE    ,.+2            ;.ASCIZ <15><12>"END OF TEST"
2078  011350  000000                            HALT
```

```
 2079  011352  012767  000320  167420  GO.NE1: MOV     #320,ICNT       ;NIETHER COMPUTER ENTRY - FIRST
 2080
 2081                                  ;*************************************************************************
 2082                                  ;TEST 321                 POWER FAIL TEST - PF - GOES DOWN - BIT13 NOT SET
 2083                                  ;*************************************************************************
 2084  011360  104400                  TST321: SCOPE
 2085
 2086  011362  012737  011420  000024          MOV     #1$,@#24        ;SET FOR POWER FAIL
 2087  011370  012737  000340  000026          MOV     #340,@#26       ;LOCK IT UP
 2088  011376  012777  011462  003440          MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
 2089  011404  012777  000502  003412          MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
 2090  011412  000001                          WAIT                    ;WAIT FOR INTERUPT
 2091  011414  000000                          HALT                    ;RETURN FROM TRAP
 2092  011416  000776                          BR      .-2             ;LOCK IT UP
 2093  011420  022626                  1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
 2094  011422  012777  011442  003562          MOV     #3$,@PUVEC$     ;RESET UP VECTOR
 2095  011430  017746  003370                  MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
 2096  011434  010667  003550                  MOV     SP,SAVE6        ;SAVE THE SP
 2097  011440  000000                          HALT                    ;WAIT FOR POWER FAIL
 2098  011442  016706  003542          3$:     MOV     SAVE6,SP        ;RESTORE THE SP
 2099  011446  012600                          MOV     (6)+,R0         ;GET FOR TYPING
 2100  011450  022700  000706                  CMP     #706,R0         ;CHECK THE WCSRA
 2101  011454  001401                          BEQ     .+4             ;SKIP IF OK
 2102  011456  104000                          HLT                     ;WCSRA NOT 706 AFTER PF
 2103  011460  000404                          BR      4$              ;SKIP WINDOW AREA
 2104  011462  022626                  2$:     CMP     (6)+,(6)+       ;CLEAR STACK
 2105  011464  017700  003334                  MOV     @WCSRA,R0       ;GET FOR TYPING
 2106  011470  104000                          HLT                     ;SHOULD NOT HAVE TRAPPED
 2107  011472  012737  000026  000024  4$:     MOV     #26,@#24        ;RESET PF VECTOR
 2108  011500  005037  177776                  CLR     @#PS            ;LOWER PROCESSOR STATUS
```

```
                                ;*****************************************************************************
                                ;TEST 322                POWER FAIL TEST - NO PF - STAYS UP - BIT13 SET
                                ;*****************************************************************************
011504  104400                  TST322: SCOPE

011506  012737  011544  000024          MOV     #1$,@#24        ;SET FOR POWER FAIL
011514  012737  000340  000026          MOV     #340,@#26       ;LOCK IT UP
011522  012777  011600  003314          MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
011530  012777  000502  003266          MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
011536  000001                          WAIT                    ;WAIT FOR INTERUPT
011540  000000                          HALT                    ;RETURN FROM TRAP
011542  000776                          BR      .-2             ;LOCK IT UP
011544  022626                  1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
011546  012777  011566  003436          MOV     #3$,@PUVEC$     ;RESET UP VECTOR
011554  017746  003244                  MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
011560  010667  003424                  MOV     SP,SAVE6        ;SAVE THE SP
011564  000000                          HALT                    ;WAIT FOR POWER FAIL
011566  016706  003416          3$:     MOV     SAVE6,SP        ;RESTORE THE SP
011572  012600                          MOV     (6)+,R0         ;GET FOR TYPING
011574  104000                          HLT                     ;DID NOT POWER FAIL
011576  000407                          BR      4$              ;SKIP WINDOW AREA
011600  022626                  2$:     CMP     (6)+,(6)+       ;CLEAR STACK
011602  017700  003216                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
011606  022700  120706                  CMP     #120706,R0      ;IS @WCSRA = 120706?
011612  001401                          BEQ     .+4             ;SKIP IF OK
011614  104000                          HLT                     ;@WCSRA NOT 120706
011616  012737  000026  000024  4$:     MOV     #26,@#24        ;RESET PF VECTOR
011624  012737  000340  177776          MOV     #340,@#PS       ;RAISE PROCESSOR STATUS TO 7
```

```
2137                                    ;*****************************************************************************
2138                                    ;TEST 323                POWER FAIL TEST - PF - GOES DOWN - BIT13 SET
2139                                    ;*****************************************************************************
2140  011632  104400                    TST323: SCOPE
2141
2142  011634  012737  011672  000024            MOV     #1$,@#24        ;SET FOR POWER FAIL
2143  011642  012737  000340  000026            MOV     #340,@#26       ;LOCK IT UP
2144  011650  012777  011734  003166            MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
2145  011656  012777  000502  003140            MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
2146  011664  000001                            WAIT                    ;WAIT FOR INTERUPT
2147  011666  000000                            HALT                    ;RETURN FROM TRAP
2148  011670  000776                            BR      .-2             ;LOCK IT UP
2149  011672  022626                    1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
2150  011674  012777  011714  003310            MOV     #3$,@PUVEC$     ;RESET UP VECTOR
2151  011702  017746  003116                    MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
2152  011706  010667  003276                    MOV     SP,SAVE6        ;SAVE THE SP
2153  011712  000000                            HALT                    ;WAIT FOR POWER FAIL
2154  011714  016706  003270            3$:     MOV     SAVE6,SP        ;RESTORE THE SP
2155  011720  012600                            MOV     (6)+,R0         ;GET FOR TYPING
2156  011722  022700  120706                    CMP     #120706,R0      ;CHECK THE WCSRA
2157  011726  001401                            BEQ     .+4             ;SKIP IF OK
2158  011730  104000                            HLT                     ;WCSRA NOT 120706 AFTER PF
2159  011732  000404                            BR      4$              ;SKIP WINDOW AREA
2160  011734  022626                    2$:     CMP     (6)+,(6)+       ;CLEAR STACK
2161  011736  017700  003062                    MOV     @WCSRA,R0       ;GET FOR TYPING
2162  011742  104000                            HLT                     ;SHOULD NOT HAVE TRAPPED
2163  011744  012737  000026  000024  4$:      MOV     #26,@#24        ;RESET PF VECTOR
2164
2165  011752  000004  011756                    TYPE    ,.+2            ;.ASCIZ <15><12>"END OF TEST"
2166  011774  000000                            HALT
```

```
2167  011776  012767  000330  166774  GO.NE2: MOV     #330,ICNT       ;NEITHER COMPUTER ENTRY - SECOND
2168
2169                                  ;*************************************************************************
2170                                  ;TEST 331                 POWER FAIL TEST - NO PF - STAYS UP - BIT13 SET
2171                                  ;*************************************************************************
2172  012004  104400                  TST331: SCOPE
2173
2174  012006  012737  012044  000024          MOV     #1$,@#24        ;SET FOR POWER FAIL
2175  012014  012737  000340  000026          MOV     #340,@#26       ;LOCK IT UP
2176  012022  012777  012100  003014          MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
2177  012030  012777  000502  002766          MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
2178  012036  000001                         WAIT                    ;WAIT FOR INTERUPT
2179  012040  000000                         HALT                    ;RETURN FROM TRAP
2180  012042  000776                         BR      .-2             ;LOCK IT UP
2181  012044  022626                  1$:    CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
2182  012046  012777  012066  003136          MOV     #3$,@PUVEC$     ;RESET UP VECTOR
2183  012054  017746  002744                  MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
2184  012060  010667  003124                  MOV     SP,SAVE6        ;SAVE THE SP
2185  012064  000000                         HALT                    ;WAIT FOR POWER FAIL
2186  012066  016706  003116          3$:    MOV     SAVE6,SP        ;RESTORE THE SP
2187  012072  012600                         MOV     (6)+,R0         ;GET FOR TYPING
2188  012074  104000                         HLT                     ;DID NOT POWER FAIL
2189  012076  000407                         BR      4$              ;SKIP WINDOW AREA
2190  012100  022626                  2$:    CMP     (6)+,(6)+       ;CLEAR STACK
2191  012102  017700  002716                  MOV     @WCSRA,R0       ;GET @WCSRA FOR TYPING
2192  012106  022700  120706                  CMP     #120706,R0      ;IS @WCSRA = 120706?
2193  012112  001401                         BEQ     .+4             ;SKIP IF OK
2194  012114  104000                         HLT                     ;@WCSRA NOT 120706
2195  012116  012737  000026  000024  4$:    MOV     #26,@#24        ;RESET PF VECTOR
```

```
2196                                    ;*****************************************************************
2197                                    ;TEST 332                POWER FAIL TEST - PF - GOES DOWN - BIT13 NOT SET
2198                                    ;*****************************************************************
2199  012124  104400                    TST332: SCOPE
2200
2201  012126  012737  012164  000024            MOV     #1$,@#24        ;SET FOR POWER FAIL
2202  012134  012737  000340  000026            MOV     #340,@#26       ;LOCK IT UP
2203  012142  012777  012226  002674            MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
2204  012150  012777  000502  002646            MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
2205  012156  000001                            WAIT                    ;WAIT FOR INTERUPT
2206  012160  000000                            HALT                    ;RETURN FROM TRAP
2207  012162  000776                            BR      .-2             ;LOCK IT UP
2208  012164  022626                    1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
2209  012166  012777  012206  003016            MOV     #3$,@PUVEC$     ;RESET UP VECTOR
2210  012174  017746  002624                    MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
2211  012200  010667  003004                    MOV     SP,SAVE6        ;SAVE THE SP
2212  012204  000000                            HALT                    ;WAIT FOR POWER FAIL
2213  012206  016706  002776            3$:     MOV     SAVE6,SP        ;RESTORE THE SP
2214  012212  012600                            MOV     (6)+,R0         ;GET FOR TYPING
2215  012214  022700  000706                    CMP     #706,R0         ;CHECK THE WCSRA
2216  012220  001401                            BEQ     .+4             ;SKIP IF OK
2217  012222  104000                            HLT                     ;WCSRA NOT 706 AFTER PF
2218  012224  000404                            BR      4$              ;SKIP WINDOW AREA
2219  012226  022626                    2$:     CMP     (6)+,(6)+       ;CLEAR STACK
2220  012230  017700  002570                    MOV     @WCSRA,R0       ;GET FOR TYPING
2221  012234  104000                            HLT                     ;SHOULD NOT HAVE TRAPPED
2222  012236  012737  000026  000024    4$:     MOV     #26,@#24        ;RESET PF VECTOR
```

```
2223                                    ;*****************************************************************
2224                                    ;TEST 333               POWER FAIL TEST - PF - GOES DOWN - BIT13 SET
2225                                    ;*****************************************************************
2226  012244  104400                    TST333: SCOPE
2227
2228  012246  012737  012304  000024            MOV     #1$,@#24        ;SET FOR POWER FAIL
2229  012254  012737  000340  000026            MOV     #340,@#26       ;LOCK IT UP
2230  012262  012777  012346  002554            MOV     #2$,@WVECA      ;SET FOR WINDOW TRAP
2231  012270  012777  000502  002526            MOV     #502,@WCSRA     ;OPEN WINDOW AND I/E
2232  012276  000001                            WAIT                    ;WAIT FOR INTERUPT
2233  012300  000000                            HALT                    ;RETUPN FROM TRAP
2234  012302  000776                            BR      .-2             ;LOCK IT UP
2235  012304  022626                    1$:     CMP     (6)+,(6)+       ;CLEAR THE STACK FROM THE PF TRAP
2236  012306  012777  012326  002676            MOV     #3$,@PUVEC$     ;RESET UP VECTOR
2237  012314  017746  002504                    MOV     @WCSRA,-(6)     ;SAVE WCSRA ON THE STACK
2238  012320  010667  002664                    MOV     SP,SAVE6        ;SAVE THE SP
2239  012324  000000                            HALT                    ;WAIT FOR POWER FAIL
2240  012326  016706  002656            3$:     MOV     SAVE6,SP        ;RESTORE THE SP
2241  012332  012600                            MOV     (6)+,R0         ;GET FOR TYPING
2242  012334  022700  120706                    CMP     #120706,R0      ;CHECK THE WCSRA
2243  012340  001401                            BEQ     .+4             ;SKIP IF OK
2244  012342  104000                            HLT                     ;WCSRA NOT 120706 AFTER PF
2245  012344  000404                            BR      4$              ;SKIP WINDOW AREA
2246  012346  022626                    2$:     CMP     (6)+,(6)+       ;CLEAR STACK
2247  012350  017700  002450                    MOV     @WCSRA,R0       ;GET FOR TYPING
2248  012354  104000                            HLT                     ;SHOULD NOT HAVE TRAPPED
2249  012356  012737  000026  000024    4$:     MOV     #26,@#24        ;RESET PF VECTOR
2250
2251  012364  000004  012370                    TYPE    ,.+2            ;.ASCIZ <15><12>"END OF TEST"
2252  012406  000000                            HALT
```

```
2253                                        ;       $SCOPE          SCOPE LOOP HANDLER
2254
2255                                        ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
2256                                        ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
2257
2258                                        ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
2259                                        ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
2260
2261  012410  032737  040000  177570  TRAP$:  BIT     #SW14,@#SWR     ;LOOP ON TEST?
2262  012416  001047                    BNE     KIT$            ;LOOP ON TEST IS SET
2263  012420  000416                    BR      3$              ;SKIP - NOP FOR XOR TESTER
2264  012422  013746  000004            MOV     @#4,-(6)        ;PUSH @#4 ON STACK
2265  012426  012737  012446  000004    MOV     #4$,@#4         ;SET FOR TIMEOUT
2266  012434  005737  177060            TST     @#177060        ;ERROR ON XOR?
2267  012440  012637  000004            MOV     (6)+,@#4        ;POP STACK INTO @#4
2268  012444  000422                    BR      SVLAD$          ;NO ERROR - GO TO NEXT TEST
2269  012446  022626              4$:   CMP     (6)+,(6)+       ;CLEAR STACK
2270  012450  012637  000004            MOV     (6)+,@#4        ;POP STACK INTO @#4
2271  012454  000430                    BR      KIT$            ;ERROR - LOOP ON TEST
2272  012456  032737  004000  177570  3$:     BIT     #SW11,@#SWR     ;KILL ITERATIONS
2273  012464  001012                    BNE     SVLAD$          ;YES - KILL ITERATIONS
2274  012466  105767  166307            TSTB    ICNT+1          ;FIRST ONE?
2275  012472  001404                    BEQ     2$              ;BRANCH IF FIRST
2276  012474  126767  000126  166277    CMPB    TIMES,ICNT+1    ;DONE?
2277  012502  001015                    BNE     KIT$            ;BRANCH IF NOT
2278  012504  112767  000001  166267  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
2279  012512  105267  166262          SVLAD$: INCB    ICNT            ;COUNT TEST NUMBERS
2280  012516  011667  166266                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
2281  012522  016737  166252  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
2282  012530  004767  000034                  JSR     PC,$TRACE
2283  012534  000002                          RTI                     ;RETURN
2284
2285  012536  105267  166237          KIT$:   INCB    ICNT+1          ;INC THE ITERATION COUNT
2286  012542  016737  166232  177570  OVER$:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
2287  012550  004767  000014                  JSR     PC,$TRACE
2288  012554  005767  166230                  TST     LAD             ;FIRST ONE?
2289  012560  001754                          BEQ     SVLAD$          ;YES
2290  012562  016716  166222                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
2291  012566  000002                          RTI                     ;FIXES PS
2292
2293  012570  032737  000400  177570  $TRACE: BIT     #SW8,@#SWR      ;CHECK BIT 8
2294  012576  001001                          BNE     .+4             ;SKIP IF TRACE
2295  012600  000207                          RTS     PC              ;RETURN
2296  012602  000004  012606                  TYPE    .+2             ;.ASCIZ <15><12>"←"
2297  012612  116705  166162                  MOVB    ICNT,TTY        ;GET BYTE FOR TYPING
2298  012616  042705  177400                  BIC     #177400,TTY     ;CLEAR JUNK
2299  012622  000167  001760                  JMP     PRINTS          ;TYPE IT
2300
2301  012626  000001                  TIMES:  1                       ;RUN 1 TIMES
```

```
2302  012630  012737  000340  000006  SETUP:  MOV     #340,@#6        ;LOCK UP 4
2303  012636  012737  010402  000020          MOV     #IOT$,@#20      ;SET UP IOT VECTOR
2304  012644  012737  000340  000022          MOV     #340,@#22       ;LOCK UP
2305  012652  012737  012410  000034          MOV     #TRAP$,@#34     ;SET TRAP VECTOR
2306  012660  012737  000340  000036          MOV     #340,@#36       ;LOCK UP
2307  012666  012737  014416  000030          MOV     #EMT$,@#30      ;SET EMT VECTOR
2308  012674  012737  000340  000032          MOV     #340,@#32       ;LOCK UP
2309  012702  005267  002300                  INC     ONCE            ;CHECK ONCE ONLY SWITCH
2310  012706  001402                          BEQ     .+6             ;SKIP JMP
2311  012710  000167  001364                  JMP     SET1            ;SKIP IF NOT FIRST TIME
2312  012714  000004  012720                  TYPE    ,.+2            ;.ASCIZ <15><12><12>"MAINDEC-11-DZDAA-A"<15><12><12>
2313  012752  012737  012766  000004  1$:     MOV     #7$,@#4         ;SET FOR TRAPPING
2314  012760  005777  002040                  TST     @WCSRA          ;IS THE ADDRESS LEAGAL
2315  012764  000440                          BR      SET2            ;YES
2316  012766  022626                  7$:     CMP     (6)+,(6)+       ;CLEAR STACK
2317  012770  012767  164000  002026          MOV     #164000,WCSRA   ;SET WCSRA FOR SCANNING
2318  012776  012701  000004                  MOV     #4,R1           ;SET COUNT
2319  013002  012737  013016  000004          MOV     #5$,@#4         ;SET FOR TIMEOUT
2320  013010  005777  002010          6$:     TST     @WCSRA          ;IS IT THERE?
2321  013014  000424                          BR      SET2            ;YES - GET OUT
2322  013016  022626                  5$:     CMP     (6)+,(6)+       ;CLEAR STACK
2323  013020  062767  000020  001776          ADD     #20,WCSRA       ;BUMP ADDRESS
2324  013026  005301                          DEC     R1              ;DEC COUNT
2325  013030  001367                          BNE     6$              ;LOOP UNTIL FOUND
2326  013032  000004  013036                  TYPE    ,.+2            ;.ASCIZ "WCSRA NOT FOUND"<15><12><12>
2327  013062  000000                          HALT                    ;NO WCSRA
2328  013064  000732                          BR      1$              ;TRY IT AGAIN
2329  013066                          SET2:
2330  013066  000004  013072                  TYPE    ,.+2            ;.ASCIZ "WCSRA = "
2331  013104  016705  001714                  MOV     WCSRA,TTY       ;TYPE WCSRA IN OCTAL
2332  013110  004767  001462                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
2333  013114  000004  013120                  TYPE    ,.+2            ;.ASCIZ "  "
2334  013124  000402                          BR      32$             ;SKIP
2335  013126  000000                  31$:    HALT                    ;WINDOW ADDRESS TIMED OUT
2336  013130  022626                          CMP     (6)+,(6)+       ;CLEAR STACK
2337  013132  012737  013126  000004  32$:    MOV     #31$,@#4        ;SET 4 FOR TIMEOUT
2338  013140  016700  001660                  MOV     WCSRA,R0        ;GET THE FIRST ADD
2339  013144  012701  015026                  MOV     #WDBRA,R1       ;GET TABLE ADDRESS
2340  013150  005720                  60$:    TST     (0)+            ;ADD 2
2341  013152  010021                          MOV     R0,(1)+         ;GET THE DATA
2342  013154  022701  015044                  CMP     #WVECA,R1       ;END?
2343  013160  001373                          BNE     60$             ;LOOP UNTIL DONE
2344  013162  012737  013172  000004          MOV     #20$,@#4        ;SET TIMEOUT RETURN
2345  013170  000402                          BR      21$             ;SKIP IT
2346  013172  000000                  20$:    HALT                    ;WINDOW IS NOT AT THIS ADDRESS
2347  013174  022626                          CMP     (6)+,(6)+       ;CLEAR THE STACK
2348  013176  012777  177777  001630  21$:    MOV     #-1,@WRARA      ;FIND THE WINDOW SIZE
2349  013204  017700  001624                  MOV     @WRARA,R0       ;SAVE THE SIZE
2350  013210  010067  001664                  MOV     R0,SIZE         ;SAVE THE SIZE FOR POSTERITY
2351  013214  005400                          NEG     R0              ;BUMP IT TO 1 BIT ONLY
2352  013216  000300                          SWAB    R0              ;GET UPPER BITS
2353  013220  006200                          ASR     R0              ;MOVE IT INTO POSITION
2354  013222  005001                          CLR     R1              ;INIT FOR OFFSET
2355  013224  005700                  54$:    TST     R0              ;IS IT 0
2356  013226  001404                          BEQ     43$             ;YES - GET OUT
2357  013230  006200                          ASR     R0              ;COUNT BITS
```

```
2358  013232  062701  000002                      ADD     #2,R1           ;KEEP COUNT
2359  013236  000772                              BR      54$             ;LOOP
2360  013240  016167  015106  001734  43$:        MOV     WIND(1),SIZ     ;GET THE SIZE
2361  013246  010167  001726                      MOV     R1,OFFSET       ;SAVE OFFSET
2362  013252  000004  015202                      TYPE    ,SIZ            ;TYPE THE WINDOW SIZE
2363  013256  000004  013262                      TYPE    ,.+2            ;.ASCIZ "K WINDOW AT "
2364  013300  017700  001532                      MOV     @WADRAX,R0      ;GET WINDOW ADDRESS
2365  013304  005067  001666                      CLR     MM              ;CLEAR MEMORY MANAGEMENT FLAG
2366  013310  005067  001506                      CLR     MX              ;GET READY TO TYPE
2367  013314  012701  000005                      MOV     #5,R1           ;INIT COUNT
2368  013320  006300                      SET3:   ASL     R0              ;MOVE IT LEFT
2369  013322  006167  001474                      ROL     MX              ;PICK UP DROPPED BIT
2370  013326  005301                              DEC     R1              ;COUNT IT
2371  013330  001373                              BNE     SET3            ;LOOP UNTIL 0
2372  013332  017700  001500                      MOV     @WADRAX,R0      ;GET AGAIN
2373  013336  006300                              ASL     R0              ;GET FOR
2374  013340  006300                              ASL     R0              ;TYPING
2375  013342  010067  001502                      MOV     R0,WADRA        ;SAVE THE WINDOW ADDRESS
2376  013346  010005                              MOV     R0,TTY          ;TYPE R0 WITH MX AS 18 BIT ADDRESS
2377  013350  004767  001256                      JSR     PC,PRINTA       ;GO TO ADDRESS PRINTER
2378  013354  012737  013550  000004              MOV     #34$,@#4        ;SET TIME OUT TRAP
2379  013362  005737  177572                      TST     @#SR0           ;CHECK FOR MM
2380  013366  005067  156746                      CLR     KIPAR0          ;MAP TO 0
2381  013372  012767  077406  156700              MOV     #77406,KIPDR0   ;READ/WRITE 4K
2382  013400  012702  000005                      MOV     #5,R2           ;SET THE COUNT
2383  013404  012701  172304                      MOV     #KIPDR2,R1      ;GET THE ADDRESS
2384  013410  012721  077406          47$:        MOV     #77406,(1)+     ;SET THE DPR
2385  013414  005302                              DEC     R2              ;CHECK COUNT
2386  013416  001374                              BNE     47$             ;LOOP
2387  013420  012767  007600  156730              MOV     #7600,KIPAR7    ;MAP TO  BANK 37
2388  013426  012767  077406  156662              MOV     #77406,KIPDR7   ;READ/WRITE 4K
2389  013434  005267  001444                      INC     FLAG            ;SET THE FLAG
2390  013440  017700  001372                      MOV     @WADRAX,R0      ;GET ADDRESS
2391  013444  006000                              ROR     R0              ;MAKE
2392  013446  006000                              ROR     R0              ;IT
2393  013450  006000                              ROR     R0              ;PAGE
2394  013452  006000                              ROR     R0              ;ADDRESS
2395  013454  042700  170000                      BIC     #170000,R0      ;CLEAR JUNK
2396  013460  010067  156660                      MOV     R0,KIPAR2       ;SET IT
2397  013464  012702  000004                      MOV     #4,R2           ;SET THE COUNT
2398  013470  012701  172346                      MOV     #KIPAR3,R1      ;GET ADDRESS
2399  013474  062700  000200          57$:        ADD     #200,R0         ;GO TO NEXT
2400  013500  010021                              MOV     R0,(1)+         ;SET IN THE ADDRESS
2401  013502  005302                              DEC     R2              ;COUNT
2402  013504  001373                              BNE     57$             ;LOOP
2403  013506  012767  040000  001334              MOV     #40000,WADRA    ;FUDGE ADDRESS
2404  013514  016767  156564  156560              MOV     KIPDR2,KIPDR1
2405  013522  016701  156616                      MOV     KIPAR2,R1
2406  013526  016702  001446                      MOV     OFFSET,R2
2407  013532  066201  015160                      ADD     ADR3(2),R1
2408  013536  010167  156600                      MOV     R1,KIPAR1
2409  013542  010667  001430                      MOV     SP,MM
2410  013546  000401                              BR      50$
2411  013550  022626                      34$:    CMP     (6)+,(6)+       ;RESTORE STACK
2412  013552  012737  000006  000004  50$:        MOV     #6,@#4          ;RESET 4
2413  013560  017700  001254                      MOV     @WVARA,R0       ;GET VECTOR ADDRESS
```

```
2414  013564  012737  000006  000004          MOV     #6,@#4          ;RESET 4
2415  013572  000004  013576                  TYPE    ,.+2            ;.ASCIZ "   VECTOR = "
2416  013614  022700  001000                  CMP     #1000,R0        ;CHECK FOR UNDER 1000
2417  013620  002403                          BLT     25$             ;GO TO ERROR ROUTINE
2418  013622  022700  000136                  CMP     #136,R0         ;CHECK FOR OVER 136
2419  013626  002403                          BLT     SET4            ;SKIP IF IN RANGE
2420  013630  000000                  25$:    HALT                    ;VECTOR IS NOT BETWEEN 140-776
2421  013632  000167  000442                  JMP     SET1            ;GET OUT IF CONTINUED
2422  013636  010067  001202          SET4:   MOV     R0,WVECA        ;SAVE THE VECTOR ADDRESS
2423  013642  005720                          TST     (0)+            ;ADD 2
2424  013644  010067  001176                  MOV     R0,WVECA+2      ;SET UP STATUS
2425  013650  017705  001164                  MOV     @WVARA,TTY      ;TYPE @WVARA IN OCTAL
2426  013654  004767  000726                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
2427 .013660  000004  013664                  TYPE    ,.+2            ;.ASCIZ <15><12>
2428  013670  022716  001024                  CMP     #ONEB1,(6)      ;ONLY 1 ADDRESS NEEDED?
2429  013674  001402                          BEQ     30$             ;NO - CONTINUE
2430  013676  000167  000364                  JMP     SET5
2431  013702  016700  001116          30$:    MOV     WCSRA,R0        ;GET STARTING ADDR OF "A" REGISTERS
2432  013706  062700  000020                  ADD     #20,R0          ;DETERMINE START OF "B" REGISTERS
2433  013712  042700  000100                  BIC     #100,R0         ;CLEAR UNWANTED CARRY
2434  013716  010067  001130                  MOV     R0,WCSRB        ;SAVE IT
2435  013722  000004  013726                  TYPE    ,.+2            ;.ASCIZ /WCSRB = /
2436  013740  016705  001106                  MOV     WCSRB,TTY       ;TYPE WCSRB IN OCTAL
2437  013744  004767  000626                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
2438  013750  000004  013754                  TYPE    ,.+2            ;.ASCIZ /   /
2439  013760  000402                          BR      10$             ;SKIP IT
2440  013762  000000                  11$:    HALT                    ;"B" WINDOW ADDRESS TIMED OUT
2441  013764  022626                          CMP     (6)+,(6)+       ;RESTORE STACK
2442  013766  012737  013762  000004  10$:    MOV     #11$,@#4        ;SET 4 FOR TIMEOUT
2443  013774  016700  001052                  MOV     WCSRB,R0        ;GET THE FIRST ADDR
2444  014000  012701  015054                  MOV     #WDBRBX,R1      ;GET TABLE ADDR
2445  014004  005720                  12$:    TST     (0)+            ;ADD 2
2446  014006  010021                          MOV     R0,(1)+         ;LOAD "B" ADDRESS
2447  014010  022701  015072                  CMP     #WVECB,R1       ;END?
2448  014014  001373                          BNE     12$             ;LOOP UNTIL DONE
2449  014016  012737  014026  000004          MOV     #13$,@#4        ;SET TIMEOUT RETURN
2450  014024  000402                          BR      14$             ;SKIP IT
2451  014026  000000                  13$:    HALT                    ;WINDOW IS NOT AT THIS ADDRESS
2452  014030  022626                          CMP     (6)+,(6)+       ;RESTORE STACK
2453  014032  012777  177777  001022  14$:    MOV     #-1,@WRARB      ;FIND THE WINDOW SIZE
2454  014040  017700  001016                  MOV     @WRARB,R0       ;SAVE THE SIZE
2455  014044  005400                          NEG     R0              ;BUMP   IT TO ONE BIT ONLY
2456  014046  000300                          SWAB    R0              ;GET UPPER BITS
2457  014050  006200                          ASR     R0              ;MOVE IT INTO POSITION
2458  014052  005001                          CLR     R1              ;INIT FOR OFFSET
2459  014054  005700                  SET6:   TST     R0              ;IS TI 0?
2460  014056  001404                          BEQ     16$             ;YES - GET OUT
2461  014060  006200                          ASR     R0              ;COUNT BITS
2462  014062  062701  000002                  ADD     #2,R1           ;KEEP COUNT
2463  014066  000772                          BR      SET6
2464  014070  016167  015106  001 004  16$:    MOV     WIND(1),SIZ     ;GET THE SIZE
2465  014076  000004  015202                  TYPE    ,SIZ            ;TYPE THE WINDOW SIZE
2466  014102  000004  014106                  TYPE    ,.+2            ;.ASCIZ /K WINDOW AT /
2467  014124  017700  000734                  MOV     @WADRBX,R0      ;GET WINDOW ADDR
2468  014130  005067  000666                  CLR     MX              ;GET READY TO TYPE
2469  014134  012701  000005                  MOV     #5,R1           ;INIT COUNT
```

```
2470  014140  006300                    17$:    ASL     R0              ;MOVE IT LEFT
2471  014142  006167  000654                    ROL     MX              ;PICK UP DROPPED BIT
2472  014146  005301                    DEC     R1              ;COUNT IT
2473  014150  001373                    BNE     17$             ;LOOP UNTIL 0
2474  014152  017700  000706                    MOV     @WADRBX,R0      ;GET ADDRESS AGAIN
2475  014156  006300                    ASL     R0              ;POSITION FOR
2476  014160  006300                    ASL     R0              ;TYPING
2477  014162  010005                    MOV     R0,TTY          ;TYPE R0 WITH MX AS 18 BIT ADDRESS
2478  014164  004767  000442                    JSR     PC,PRINTA       ;GO TO ADDRESS PRINTER
2479  014170  017700  000672                    MOV     @WVARB,R0       ;GET THE B VECTOR
2480  014174  000004  014200                    TYPE    .+2             ;.ASCIZ /   VECTOR = /
2481  014216  017705  000644                    MOV     @WVARB,TTY      ;TYPE @WVARB IN OCTAL
2482  014222  004767  000360                    JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
2483  014226  000004  014232                    TYPE    .+2             ;.ASCIZ <15><12>
2484  014236  022700  001000                    CMP     #1000,R0        ;CHECK FOR UNDER 1000
2485  014242  002403                    BLT     27$             ;ERROR IF OVER 1000
2486  014244  022700  000136                    CMP     #136,R0         ;CHECK FOR UNDER 136
2487  014250  002401                    BLT     28$             ;SKIP IF UNDER 140
2488  014252  000000                    27$:    HALT                    ;ADDRESS NOT 140-776
2489  014254  010067  000612                    28$:    MOV     R0,WVECB        ;SAVE THE ADDRESS
2490  014260  005720                    TST     (0)+            ;ADD 2
2491  014262  010067  000606                    MOV     R0,WVECB+2      ;SAVE THE STATUS ADDRESS
2492  014266  012737  000006  000004    SET5:   MOV     #6,@#4          ;RESTORE 4
2493  014274  012706  000476                    MOV     #476,SP         ;RESET SP
2494  014300  005067  164504            SET1:   CLR     LAD             ;CLEAR THE LOOP ADDRESS
2495  014304  005067  164470                    CLR     ICNT            ;CLEAR THE COUNT
2496  014310  016777  000532  000526            MOV     WVECA+2,@WVECA  ;LOCK OUT DA11 TRAPS
2497  014316  005077  000524                    CLR     @WVECA+2        ;HALT ADDRESS
2498  014322  016777  000546  000542            MOV     WVECB+2,@WVECB  ;LOCK OUT OTHER ADDRESS
2499  014330  005077  000540                    CLR     @WVECB+2        ;HALT ADDRESS
2500  014334  012737  000006  000004            MOV     #6,@#4          ;RESET 4
2501  014342  005767  000536                    TST     FLAG            ;MM IN USE?
2502  014346  001403                    BEQ     3$              ;NO - GET OUT
2503  014350  012737  000001  177572            MOV     #1,@#SRC        ;TURN IT ON
2504  014356  000207                    3$:     RTS     PC              ;RETURN
2505
2506
2507  014360  012704  000020            TIMO:   MOV     #20,R4          ;SET COUNT FOR DOUBLE LOOP
2508  014364  005005                    1$:     CLR     R5              ;ZERO COUNT
2509  014366  032777  010000  000430    2$:     BIT     #10000,@WCSRA   ;CHECK FOR NEW DATA B
2510  014374  001401                    BEQ     3$              ;SKIP IF NOT SET
2511  014376  000207                    4$:     RTS     PC              ;RETURN
2512  014400  005205                    3$:     INC     R5              ;BUMP COUNT
2513  014402  001371                    BNE     2$              ;LOOP IF NO TIMEOUT
2514  014404  005304                    DEC     R4              ;DEC OTHER COUNT
2515  014406  001366                    BNE     1$              ;LOOP IF NO TIMEOUT
2516  014410  011600                    MOV     (6),R0          ;GET THE ADDRESS
2517  014412  104000                    HLT                     ;NO TIMEOUT
2518  014414  000000                    HALT
```

```
                                        ;       $HLT            ERROR TYPEOUT HANDLER
                                        ;
                                        ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
                                        ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
                                        ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
                                        ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUEMENT
                                        ;(HLT+3) WILL BE PLACED IN "HLTCT$:" FOR ADITIONAL TYPECUTS.
                                        ;
014416  032737  002000  177570  EMT$:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
014424  001402                          BEQ     1$              ;NO - SKIP
014426  000004  000007                  TYPE    ,BELL           ;RING BELL
014432  005267  164344          1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
014436  032737  020000  177570          BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
014444  001026                          BNE     2$              ;SKIP TYPEOUTS
014446  000004  014452                  TYPE    .+2             ;.ASCIZ <15><12>
014456  011667  164330                  MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
014462  162767  000002  164322          SUB     #2,HLTADR       ;FUDGE ADDRESS
014470  117767  164316  000036          MOVB    @HLTADR,HLTCT$  ;GET HLT ARGUEMENT
014476  016705  164310                  MOV     HLTADR,TTY      ;TYPE HLTADR IN OCTAL
014502  004767  000070                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
014506  000004  014512                  TYPE    .+2             ;.ASCIZ " "
014516  004767  000014                  JSR     PC,ROUTIN       ;GO TO USER ERROR ROUTINE
014522  005737  177570          2$:     TST     @#SWR           ;HALT ON ERROR
014526  100001                          BPL     .+4             ;SKIP IF CONTINUE
014530  000000                          HALT                    ;HALT ON ERROR!
014532  000002                          RTI                     ;RETURN
                                        ;
014534  000000                  HLTCT$: 0                       ;HLT ARGUEMENT
                                        ;
014536                          ROUTIN:
014536  010005                          MOV     R0,TTY          ;TYPE R0 IN OCTAL
014540  004767  000032                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
014544  005767  177764                  TST     HLTCT$          ;CHECK FOR 2 WORD TYPE
014550  001405                          BEQ     1$              ;RETURN IF 0
014552  000004  000040                  TYPE    ,40             ;TYPE A SPACE
014556  010105                          MOV     R1,TTY          ;TYPE R1 IN OCTAL
014560  004767  000012                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
014564  000207                  1$:     RTS     PC              ;RETURN TO SCOPE
```

```
                                        :       $OCTAL          OCTAL TYPEOUT ROUTINE
                                        ;
                                        ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
                                        ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
                                        ;THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S.
                                        ;
014566  012767  170101  000202  BITYP$: MOV     #170101,.PR     ;SET BIT FLAG ANS 16. CHARACTER COUNT
014574  000411                          BR      .PTIT           ;NOW TYPE IT IN BIT FORM
014576  112767  000001  000172  PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
014604  000402                          BR      .+6             ;SKIP
014606  005067  000164          PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
014612  112767  177772  000157          MOVB    #-6,.PR+1       ;SET COUNT
014620  010446                  .PTIT:  MOV     R4,-(6)         ;SAVE R4
014622  012704  015000                  MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
014626  105014                          CLRB    (4)             ;CLEAR FIRST BYTE
014630  000432                          BR      .PRF            ;ROTATE FIRST BIT
014632  010446                  PRINTA: MOV     R4,-(6)         ;SAVE R4
014634  012704  015000                  MOV     #.PR+2,R4       ;SET UP POINTER TO OUTPUT AREA
014640  116714  000156                  MOVB    MX,(4)          ;MX CONTAINS UPPER 5 BITS
014644  006305                          ASL     TTY             ;GET RID
014646  006305                          ASL     TTY             ;OF 3
014650  006305                          ASL     TTY             ;JUNK BITS
014652  106214                          ASRB    (4)             ;GET BIT13
014654  006005                          ROR     TTY             ;PACK IT
014656  106214                          ASRB    (4)             ;GET BIT14
014660  006005                          ROR     TTY             ;PACK IT
014662  152724  000060                  BISB    #'0,(4)+        ;MAKE IT ASCII
014666  012767  175401  000102          MOV     #175401,.PR     ;-5,,1 - 5 BYTES AND FILL
014674  105014                  .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
014676  032767  000100  000072          BIT     #100,.PR        ;BIT TYPING MODE?
014704  001004                          BNE     .PRF            ;YES - SKIP 2 ROTATES
014706  006105                          ROL     TTY             ;ROTATE BIT INTO C
014710  106114                          ROLB    (4)             ;PACK IT
014712  006105                          ROL     TTY             ;ROTATE BIT INTO C
014714  106114                          ROLB    (4)             ;PACK IT
014716  006105                  .PRF:   ROL     TTY             ;ROTATE BIT INTO C
014720  106114                          ROLB    (4)             ;PACK IT
014722  105714                          TSTB    (4)             ;IS IT ZERO?
014724  001402                          BEQ     .+6             ;SKIP INC
014726  105267  000044                  INCB    .PR             ;SET FILL SWITCH
014732  105767  000040                  TSTB    .PR             ;CHECK FILL SWITCH
014736  001402                          BEQ     .+6             ;SKIP BITSET
014740  152724  000060                  BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
014744  105267  000027                  INCB    .PR+1           ;INC COUNT
014750  001351                          BNE     .PRL            ;REPEAT
014752  022704  015000                  CMP     #.PR+2,R4       ;EMPTY BUFFER?
014756  001002                          BNE     .+6             ;SKIP IF NOT
014760  112724  000060                  MOVB    #'0,(4)+        ;LOAD 1 ZERO
014764  105014                          CLRB    (4)             ;NULL TERMINATOR
014766  000004  015000                  TYPE    .PR+2           ;TYPE IT
014772  012604                          MOV     (6)+,R4         ;RESTORE R4
014774  000207                          RTS     PC              ;RETURN
014776  000012                  .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
015022  000000                  MX:     0                       ;MEMORY EXTENSION BITS
```

```
2611  015024  164000                  WCSRA:  164000                  ;CONTROL & STATUS REG. (R/W)  -  WINDOW A
2612  015026  164002                  WDBRA:  164002                  ;OUTPUT DATA BUFFER (R/W)
2613  015030  164004                  WDBRB:  164004                  ;INPUT DATA BUFFER (R)
2614  015032  164006                  WDARA:  164006                  ;DISPLACEMENT ADDRESS REG. (R)
2615  015034  164010                  WRARA:  164010                  ;RELOCATION ADDRESS REGISTER (R/W)
2616  015036  164012                  WADRAX: 164012                  ;WINDOW ADDRESS REGISTER (R)
2617  015040  164014                  WVARA:  164014                  ;VECTOR ADDRESS REGISTER (R)
2618  015042  000000                  0
2619  015044  000300  000302          WVECA:  300,302                 ;VECTOR ADDRESS
2620  015050  020000                  WADRA:  20000                   ;WINDOW ADDRESS
2621
2622  015052  164020                  WCSRB:  164020                  ;CONTROL & STATUS REG. (R/W)  -  WINDOW B
2623  015054  164022                  WDBRBX: 164022                  ;OUTPUT DATA BUFFER (R/W)
2624  015056  164024                  WDBRAX: 164024                  ;INPUT DATA BUFFER (R)
2625  015060  164026                  WDARB:  164026                  ;DISPLACEMENT ADDRESS REG. (R)
2626  015062  164030                  WRARB:  164030                  ;RELOCATION ADDRESS REGISTER (R/W)
2627  015064  164032                  WADRBX: 164032                  ;WINDOW ADDRESS REGISTER (R)
2628  015066  164034                  WVARB:  164034                  ;VECTOR ADDRESS REGISTER (R)
2629  015070  000000                  0
2630  015072  000304  000306          WVECB:  304,306                 ;VECTOR ADDRESS FOR WINDOW B
2631  015076  040000                  WADRB:  40000                   ;WINDOW ADDRESS FOR B
2632
2633  015100  000000                  SIZE:   0                       ;WINDOW SIZE
2634  015102  000000                  DISP:   0
2635  015104  000000                  FLAG:   0                       ;MM FLAG
2636  015106  032456  030440  031040  WIND:   .ASCII  ".5 1 2 4 81632"
2637  015114  032040  034040  033061
2638  015122  031063
2639  015124  176400  176000  176000  ADR1:   176400,176000,176000,174000,170000,160000,140000
2640  015132  174000  170000  160000
2641  015140  140000
2642  015142  001000  003000  003000  ADR2:   1000,3000,3000,13000,33000,73000,173000
2643  015150  013000  033000  073000
2644  015156  173000
2645  015160  000010  000030  000030  ADR3:   10,30,30,130,330,730,1730
2646  015166  000130  000330  000730
2647  015174  001730
2648  015176  000000                  MM:     0                       ;MEMORY MANAGEMENT FLAG
2649  015200  000000                  OFFSET: 0
2650  015202  000000  000000          SIZ:    0,0
2651  015206  177777                  ONCE:   -1                      ;ONCE ONLY FLAG
2652  015210  000000                  SAVE6:  0                       ;FOR SP
2653  015212  000024  000026          PUVEC$: 24,26                   ;POWER UP VECTOR ADDRESS
2654
2655          177572                  SR0=    177572                  ;KT11 - MEMORY MANAGEMENT
2656          172340                  KIPAR0= 172340
2657          172300                  KIPDR0= 172300
2658          172342                  KIPAR1=172342
2659          172302                  KIPDR1=172302
2660          172344                  KIPAR2= 172344
2661          172304                  KIPDR2= 172304
2662          172346                  KIPAR3= 172346
2663          172306                  KIPDR3= 172306
2664          172356                  KIPAR7= 172356
2665          172316                  KIPDR7= 172316
2666
```

```
                017440                  .=      17440
                
017440  000005          OPEN:   RESET                   ;ZAP THE WORLD
017442  012706  000500          MOV     #500,SP         ;INITIALIZE THE STACK
017446  012767  177777  175532  MOV     #-1,ONCE        ;RESET
017454  012767  000777  000014  MOV     #777,BR.        ;RESET BR.
017462  004767  161074          JSR     PC,REGSCN       ;GET THE REGISTER
017466  010067  000046          MOV     R0,BR.1
017472  012710  000402          MOV     #402,(0)        ;OPEN THE WINDOW
017476  000777          BR.:    BR      .               ;WAIT
017500  000240                  NOP
017502  000240                  NOP
017504  000240                  NOP
017506  000240                  NOP
017510  000240                  NOP
017512  005077  000022          CLR     @BR.1           ;CLEAR WCSRA
017516  012701  000100          MOV     #100,R1
017522  005000          20$:    CLR     R0              ;WAIT
017524  005300          10$:    DEC     R0              ;BEFORE
017526  001376                  BNE     10$             ;STARTING
017530  005301                  DEC     R1
017532  001373                  BNE     20$
017534  000137  003410          JMP     @#BEGIN         ;SELF START
017540  000000          BR.1:   0
                
                000001          .END
```

```
ADR1    015124          1815    2639#
ADR2    015142          1358    1380    2642#
ADR3    015160          2407    2645#
BEGIN   003410           448     472     994#   2689
BELL  = 000007           434#    992    1907    2529
BITYPS  014566          2563#
BR.     017476           471#   2672#   2676#
BR.1    017540          2674#   2682#   2690#
CHKADR  000740           489#    491#    495     500#
CODE1   000700           488#   1516
DISP    015102          1836#   1851    1855#   2634#
DISPLA= 177570           433#   2281#   2286#
DONE    010330          1477    1549    1902#
EMTS    014416          2307    2527#
ERRORS  001002           505#   2530#   2546
FILCHR  010550          1938    1939    1958#
FLAG    015104          2389#   2501    2635#
GETCOR  000514           457     461#    463
GO      003422           997#   1914
GO.HST  010556           450    1961#
GO.NE1  011352           452    2079#
GO.NE2  011776           453    2167#
GO.OTH  011060           451    2021#
HLT   = 104000           429#    498     523     529     535     541     547     557     563     569     575     581     591
                         597     603     609     615     625     631     637     643     649     659     665     671     677
                         683     693     699     705     711     717     733     742     751     760     773     782     791
                         800     810     816     822     828     834     840     846     852     858     864     870     883
                         889     895     901     907     913     919     925     931     937     943     959     964     969
                         973     978     983    1016    1026    1036    1046    1057    1063    1069    1075    1091    1100
                        1109    1118    1132    1139    1146    1153    1160    1167    1174    1181    1188    1195    1202
                        1208    1222    1230    1238    1246    1260    1265    1275    1280    1290    1296    1304    1311
                        1321    1327    1333    1341    1348    1367    1374    1386    1411    1417    1448    1475    1504
                        1527    1534    1561    1567    1573    1579    1590    1596    1602    1608    1624    1633    1642
                        1651    1666    1673    1680    1687    1694    1701    1708    1715    1722    1729    1742    1750
                        1758    1766    1781    1787    1797    1848    1854    1985    1989    2011    2017    2044    2048
                        2071    2075    2102    2106    2128    2134    2158    2162    2188    2194    2217    2221    2244
                        2248    2517
HLTADR  001012           508#   2534#   2535#   2536    2537    2547
HLTCTS  014534          2536#   2546#   2551
ICNT    001000           504#   1002#   1550#   1961#   2021#   2079#   2167#   2274    2276    2278#   2279#   2281    2285#
                        2286    2297    2301    2495#
IOTS    010402          1927#   2303
IOTIS   010554          1927#   1956    1960#
KIPAR0= 172340          2380#   2656#
KIPAR1= 172342          2408#   2658#
KIPAR2= 172344          2396#   2405    2660#
KIPAR3= 172346          2398    2662#
KIPAR7= 172356          2387#   2664#
KIPDR0= 172300          2381#   2657#
KIPDR1= 172302          2404#   2659#
KIPDR2= 172304          2383    2404    2661#
KIPDR3= 172306          2663#
KIPDR7= 172316          2388#   2665#
KITS    012536          2262    2271    2277    2285#
LAD     001010           507#   2280#   2288    2290    2301    2494#
MM      015176          1353    1377    2365#   2409#   2648#
```

DZDAAA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
MX      015022          2366* 2369* 2468* 2471* 2575  2610*
N     = 000334           427*  514   518*  548   552*  582   586*  616   620*  650   654*  684   688*
                         721   725*  761   765*  801   805*  874   878*  947   951*  994*  1004  1008*
                        1048  1052* 1076  1080* 1120  1124* 1209  1213* 1247  1251* 1281  1285* 1387
                        1391* 1425  1429* 1457  1461* 1486  1490* 1508  1512* 1518  1522* 1550* 1552
                        1556* 1581  1585* 1609  1613* 1653  1657* 1731  1735* 1770  1774* 1799  1803*
                        1819  1823* 1832  1836* 1861  1865* 1874  1878* 1882  1886* 1892  1896* 1961*
                        1964  1968* 1992  1996* 2021* 2023  2027* 2050  2054* 2079* 2081  2085* 2109
                        2113* 2137  2141* 2167* 2169  2173* 2196  2200* 2223  2227*
OFFSET  01520C          1357  1379  1814  2361* 2406  2649*
ONCE    015206          2309* 2631* 2671*
ONEBUS  001014           449   510*  993  1465  1466
ONEB1   001024           512* 2428
OPEN    017440           459  2669*
OVER$   012542          2286*
PC    =%000007           443*  461   478*  494*  511*  996*  1135* 1142* 1149* 1156* 1163* 1170* 1177*
                        1184* 1191* 1198* 1215* 1218* 1226* 1234* 1242* 1253* 1256* 1268* 1271* 1287*
                        1299* 1324* 1336* 1352* 1394* 1424* 1432* 1451* 1456* 1463* 1483* 1492* 1514*
                        1516* 1526* 1540* 1661* 1668* 1675* 1682* 1689* 1696* 1703* 1710* 1717* 1724*
                        1736* 1738* 1746* 1754* 1762* 1775* 1777* 1790* 1793* 1804* 1807* 1809* 1811*
                        1813* 1824* 1837* 1842* 1866* 1879* 1887* 1897* 1900* 2282* 2287* 2295* 2332*
                        2377* 2426* 2437* 2478* 2482* 2504* 2511* 2538* 2540* 2550* 2555* 2556* 2608*
                        2673*
PCNT    001004           506* 1903* 1904*
PRINTA  014632          2377  2478  2573*
PRINTR  014576          2332  2437  2538  2550  2555  2565*
PRINTS  014606          2299  2426  2482  2567*
PS    = 177776           431*  512*  952*  997* 1126* 1524  1659* 1991* 2108* 2136*
PUVEC$  015212          1977* 2005* 2036* 2063* 2094* 2122* 2150* 2182* 2209* 2236* 2653*
QQ    = 000001           514*  548*  582*  616*  650*  684*  721*  761*  801*  874*  947*  1004* 1048*
                        1076* 1120* 1135  1136* 1142  1143* 1149  1150* 1156  1157* 1163  1164* 1170
                        1171* 1177  1178* 1184  1185* 1191  1192* 1198  1199* 1209* 1215  1216* 1218
                        1219* 1226  1227* 1234  1235* 1242  1243* 1247* 1253  1254* 1256  1257* 1268
                        1269* 1271  1272* 1281* 1287  1288* 1299  1300* 1324  1325* 1336  1337* 1352
                        1353* 1387* 1394  1395* 1424  1425* 1432  1433* 1451  1452* 1456  1457* 1463
                        1464* 1483  1484* 1486* 1492  1493* 1508* 1514  1515* 1518* 1526  1527* 1540
                        1541* 1552* 1581* 1609* 1653* 1661  1662* 1668  1669* 1675  1676* 1682  1683*
                        1689  1690* 1696  1697* 1703  1704* 1710  1711* 1717  1718* 1724  1725* 1731*
                        1736  1737* 173   1739* 1746  1747* 1754  1755* 1762  1763* 1770* 1775  1776*
                        1777  1778* 1790  1791* 1793  1794* 1799* 1804  1805* 1807  1808* 1809  1810*
                        1811  1812* 1813  1814* 1819* 1824  1825* 1832* 1837  1838* 1842  1843* 1861*
                        1866  1867* 1874* 1879  1880* 1882* 1887  1888* 1892* 1897  1898* 1900  1901*
                        1964* 1992* 2023* 2050* 2081* 2109* 2137* 2169* 2196* 2223*
R     = 000100           811*  817*  823*  829*  835*  841*  847*  853*  884*  890*  896*  902*  908*
                         914*  920*  926* 1134* 1136  1141* 1143  1148* 1150  1155* 1157  1162* 1164
                        1169* 1171  1176* 1178  1183* 1185  1190* 1663* 1667  1670* 1674  1677* 1681
                        1684* 1688  1691* 1695  1698* 1702  1705* 1709  1712* 1716  1719*
REGSCN  000562           474*  484  2673
ROUTIN  014536          2540  2548*
RO    =%000000           435*  464*  466  474*  480*  520*  521  526*  527  532*  533  538*  539
                         544*  545  554*  555  560*  561  566*  567  572*  573  578*  579  588*
                         589  594*  595  600*  601  606*  607  612*  613  622*  623  628*  629
                         634*  635  640*  641  646*  647  656*  657  662*  663  668*  669  674*
                         675  680*  681  690*  691  696*  697  702*  703  708*  709  714*  715
                         729*  730*  731  738*  739*  740  747*  748*  749  756*  757*  758  769*
                         770*  771  778*  779*  780  787*  788*  789  796*  797*  798  807*  808
```

```
                    813*  814   819*  820   825*  826   831*  832   837*  838   843*  844   849*
                    850   855*  856   861*  862   867*  868   880*  881   886*  887   892*  893
                    898*  899   904*  905   910*  911   916*  917   922*  923   928*  929   934*
                    935   940*  941   961*  962   966*  967   975*  976   980*  981   1012* 1013*
                   1014  1022* 1023* 1024  1032* 1033* 1034  1042* 1043* 1044  1054* 1055  1060*
                   1061  1066* 1067  1072* 1073  1087* 1088* 1089  1096* 1097* 1098  1105* 1106*
                   1107  1114* 1115* 1116  1129* 1130  1136* 1137  1143* 1144  1150* 1151  1157*
                   1158  1164* 1165  1171* 1172  1178* 1179  1185* 1186  1192* 1193  1199* 1200
                   1205* 1206  1219* 1220  1227* 1228  1235* 1236  1243* 1244  1257* 1258  1262*
                   1263  1272* 1273  1277* 1278  1293* 1294  1303* 1308* 1309  1318* 1319  1330*
                   1331  1340* 1345* 1346  1364* 1365  1369* 1371* 1372  1383* 1384  1408* 1409
                   1414* 1415  1445* 1446  1464* 1465* 1479  1493* 1494* 1499* 1515* 1531* 1532
                   1543* 1544* 1558* 1559  1564* 1565  1570* 1571  1576* 1577  1587* 1588  1593*
                   1594  1599* 1600  1605* 1606  1620* 1621* 1622  1629* 1630* 1631  1638* 1639*
                   1640  1647* 1648* 1649  1663* 1664  1670* 1671  1677* 1678  1684* 1685  1691*
                   1692  1698* 1699  1705* 1706  1712* 1713  1719* 1720  1726* 1727  1739* 1740
                   1747* 1748  1755* 1756  1763* 1764  1778* 1779  1784* 1785  1794* 1795  1845*
                   1846  1850* 1852  1908* 1982* 1983  1988* 2010* 2014* 2015  2041* 2042  2047*
                   2068* 2069  2074* 2099* 2100  2105* 2127* 2131* 2132  2155* 2156  2161* 2187*
                   2191* 2192  2214* 2215  2220* 2241* 2242  2247* 2338* 2341  2349* 2350  2351*
                   2352* 2353* 2355  2357* 2364* 2368* 2372* 2373* 2374* 2375  2376  2390* 2391*
                   2392* 2393* 2394* 2395* 2396  2399* 2400  2413* 2416  2418  2422  2424  2431*
                   2432* 2433* 2434  2443* 2446  2454* 2455* 2456* 2457* 2459  2461* 2467* 2470*
                   2474* 2475* 2476* 2477  2479* 2484  2486  2489  2491  2516* 2549  2674  2684*
                   2685*
R1      =%000001    436#  461*  463*  1355* 1357* 1358* 1359* 1379* 1380* 1382* 1384  1403* 1404
                   1413* 1415  1418* 1419* 1420  1441* 1442  1452* 1472* 1501* 1542* 1546* 1851*
                   1852  2318* 2324* 2339* 2342  2354* 2358* 2361  2367* 2370* 2383* 2398* 2405*
                   2407* 2408  2444* 2447  2458* 2462* 2469* 2472* 2554  2683* 2687*
R2      =%000002    437#  475*  481*  718*  719*  727   730   736   739   745   748   754   757
                    767   770   776   779   785   788   794   797   1010* 1013  1020  1023  1030
                   1033  1040  1043  1081* 1082* 1085  1088  1094  1097  1103  1106  1112  1115
                   1395* 1396* 1397* 1398* 1401* 1404* 1420  1433* 1434* 1435* 1436* 1439* 1453*
                   1467* 1468* 1469* 1470* 1471* 1479  1614* 1615* 1618  1621  1627  1630  1636
                   1639  1645  1648  2382* 2385* 2397* 2401* 2406*
R3      =%000003    438#  726*  727*  728   731   735*  736*  737   740   744*  745*  746   749
                    753*  754*  755   758   766*  767*  768   771   775*  776*  777   780   784*
                    785*  786   789   793*  794*  795   798   1019* 1020* 1021  1024  1029* 1030*
                   1031  1034  1039* 1040* 1041  1044  1084* 1085* 1086  1089  1093* 1094* 1095
                   1098  1102* 1103* 1104  1107  1111* 1112* 1113  1116  1466* 1477  1495* 1496
                   1497* 1500* 1502  1506* 1617* 1618* 1619  1622  1626* 1627* 1628  1631  1635*
                   1636* 1637  1640  1644* 1645* 1646  1649
R4      =%000004    439#  1442* 1927  1938* 1942* 1956* 2507* 2514* 2569  2570* 2573  2574* 2602
                   2607*
R5      =%000005    440#  2508* 2512*
SAVE6     015210   1979* 1981  2007* 2009  2038* 2040  2065* 2067  2096* 2098  2124* 2126  2152*
                   2154  2184* 2186  2211* 2213  2238* 2240  2652#
SCOPE = 104400      428#  517   551   585   619   653   687   724   764   804   877   950   989
                   1007  1051  1079  1123  1212  1250  1284  1390  1428  1460  1489  1511  1521
                   1555  1584  1612  1656  1734  1773  1802  1822  1835  1864  1877  1885  1895
                   1902  1967  1995  2026  2053  2084  2112  2140  2172  2199  2226
SECOND    006420   1001  1550#
SETUP     012630    511   996  2302#
SET1      014300   2311  2421  2494#
SET2      013066   2315  2321  2329#
SET3      013320   2368# 2371
```

```
SET4     013636            2419    2422*
SET5     014266            2430    2492*
SET6     014054            2459*   2463
SIZ      015202            2360*   2362    2464*   2465    2650*
SIZE     015100             718    1081    1395    1399    1433    1437    1467    1614    2350*   2633*
SP     =%000006             442*    510*    960*    974*    995*   1962*   1979   1981*   2007    2009*   2038    2040*   2065
                          2067*   2096    2098*   2124    2126*   2152    2154*   2184    2186*   2211    2213*   2238    2240*
                          2409    2493*   2670*
SR0    = 177572           2379    2503*   2655*
SVLAD$   012512           2268    2273    2279*   2289
SWR    = 177570            432*    433     468     990    1905    2261    2272    2293    2527    2531    2541
SW10   = 002000            382*    990    1905    2527
SW11   = 004000            381*   2272
SW12   = 010000            380*
SW13   = 020000            379*   2531
SW14   = 040000            378*   2261
SW15   = 100000            377*
SW7    = 000200            386*    468
SW8    = 000400            385*   2261    2293
SW9    = 001000            384*   2544
TIMES    012626           2276    2301*
TIMO     014360           1135    1142    1149    1156    1163    1170    1177    1184    1191    1198    1215    1218    1226
                          1234    1242    1253    1256    1268    1271    1287    1299    1324    1336    1352    1394    1424
                          1432    1451    1456    1463    1483    1492    1514    1526    1540    1661    1668    1675    1682
                          1689    1696    1703    1710    1717    1724    1736    1738    1746    1754    1762    1775    1777
                          1790    1793    1804    1807    1809    1811    1813    1824    1837    1842    1866    1879    1887
                          1897    1900    2507*
TRAP$    012410           2261*   2305
TST1     001032            517*
TST10    002176            764*
TST101   003454           1007*
TST102   003620           1051*
TST103   003730           1079*
TST104   004074           1123*
TST105   004510           1212*
TST106   004704           1250*
TST107   005052           1284*
TST11    002330            804*
TST110   005530           1390*
TST111   005714           1428*
TST112   006062           1460*
TST113   006166           1489*
TST114   006252           1511*
TST115   006276           1521*
TST12    002650            877*
TST13    003170            950*
TST2     001160            551*
TST201   006426           1555*
TST202   006546           1584*
TST203   006656           1612*
TST204   007022           1656*
TST205   007376           1734*
TST206   007572           1773*
TST207   007724           1802*
TST210   010026           1822*
TST211   010064           1835*
```

```
TST212  010204          1864*
TST213  010242          1877*
TST214  010262          1885*
TST215  010302          1895*
TST3    001306           585*
TST301  010570          1967*
TST302  010714          1995*
TST311  011066          2026*
TST312  011206          2053*
TST321  011360          2094*
TST322  011504          2112*
TST323  011632          2140*
TST331  012004          2172*
TST332  012124          2199*
TST333  012244          2226*
TST4    001434           619*
TST5    001562           653*
TST6    001710           687*
TST7    002044           724*
TTY    =%000005          441*  1814*  1815   1928   1929*  1930   1932   1933*  1934   1936   1944   1952*  1953*
                        1954   1955*  2297*  2298*  2331*  2376*  2425*  2436*  2477*  2481*  2537*  2549*  2554*
                        2576*  2577*  2578*  2580*  2582*  2588*  2590*  2592*
TYPE   = 000004          430*   992   1907   2019   2077   2165   2251   2296   2312   2326   2330   2333   2362
                        2363   2415   2427   2435   2438   2465   2466   2480   2483   2529   2533   2539   2553
                        2606
WADRA   015050          1289   1301   1314*  1326*  1338*  1359   1382   1403   1413   1419   1441   1464   1471
                        1493   1515   2375*  2403*  2620*
WADRAX  015036          1839   2364   2372   2390   2616*
WADRB   015076          2631*
WADRBX  015064          2467   2474   2627*
WCSRA   015024           520    525*   526    531*   532    537*   538    543*   544    806*   812*   818*   824*
                         830*   836*   842*   848*   854*   860*   866*   872*   880    886    892    898    904
                         910    916    922    928    934    940    946*   955*   957*   966    975    979*   985*
                         999   1009*  1010   1011*  1012   1021*  1022   1031*  1032   1041*  1042   1047*  1127
                        1129   1134*  1136   1141*  1143   1148*  1150   1155*  1157   1162*  1164   1169*  1171
                        1176*  1178   1183*  1185   1190*  1192   1197*  1199   1204*  1205   1214*  1217*  1225*
                        1233*  1241*  1252*  1255*  1267*  1270*  1286*  1293   1298*  1303   1308   1318   1323*
                        1330   1335*  1340   1345   1351*  1364   1369   1371   1393*  1408   1412*  1423*  1431*
                        1445   1449*  1450*  1455*  1462*  1482*  1491*  1513*  1525*  1531   1539*  1557*  1558
                        1563*  1564   1569*  1570   1575*  1576   1580*  1660*  1663   1667*  1670   1674*  1677
                        1681*  1684   1688*  1691   1695*  1698   1702*  1705   1709*  1712   1716*  1719   1723*
                        1726   1730*  1737*  1745*  1753*  1761*  1769*  1776*  1789*  1792*  1798*  1806*  1808*
                        1810*  1812*  1816*  1826*  1827   1829*  1841*  1843   1845   1849*  1858*  1868*  1869
                        1871*  1881*  1890*  1898*  1901*  1972*  1978   1988   2000*  2006   2014   2031*  2037
                        2047   2058*  2064   2074   2089*  2095   2105   2117*  2123   2131   2145*  2151   2161
                        2177*  2183   2191   2204*  2210   2220   2231*  2237   2247   2314   2317*  2320   2323*
                        2331   2338   2431   2509   2611*
WCSRB   015052           554    559*   560    565*   566    571*   572    577*   578    807    813    819    825
                         831    837    843    849    855    861    867    873*   879*   885*   891*   897*   903*
                         909*   915*   921*   927*   933*   939*   945*   956*   961    965*   971*   980    986*
                        2434*  2436   2443   2622*
WDARA   015032          1383   1414   1850   2614*
WDARB   015060          2625*
WDBRA   015026           588    593*   594    599*   600    605*   606    611*   612    661*   667*   673*   679*
                        1053*  1054   1059*  1060   1065*  1066   1071*  1072   1216*  1224*  1232*  1240*  1254*
                        1257   1269*  1272   1586*  1587   1592*  1593   1598*  1599   1604*  1605   1744*  1752*
```

DZDAAA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    1760* 1768* 1783* 1784  1791* 1794  2339  2612*
WDBRAX  015056       656   662   668   674   680  2624*
WDBRB   015030       690   696   702   708   714  1219  1227  1235  1243  1262  1277  1739  1747
                    1755  1763  1778  2613*
WDBRBX  015054       622   627*  628   633*  634   639*  640   645*  646   695*  701*  707*  713*
                    2444  2623*
WIND    015106      2360  2464  2636*
WRARA   015034       728*  729   737*  738   746*  747   755*  756   1086* 1087  1095* 1096  1104*
                    1105  1113* 1114  1119* 1392* 1430* 1619* 1620  1628* 1629  1637* 1638  1646*
                    1647  1652* 1805* 1815* 1825* 1839* 1867* 1880* 1889* 2348* 2349  2615*
WRARB   015062       768*  769   777*  778   786*  787   795*  796   2453* 2454  2626*
WVARA   015040      2413  2425  2617*
WVARB   015066      2479  2481  2628*
WVECA   015044       953*  1523* 1524* 1535* 1536* 1971* 1999* 2030* 2057* 2088* 2116* 2144* 2176*
                    2203* 2230* 2342  2422* 2424* 2496* 2497* 2619*
WVECB   015072       954*  2447  2489* 2491* 2498* 2499* 2630*
$TRACE  012570      2282  2287  2293*
.     = 017542       444*  445   446*  455*  469   486*  493   497   502*  522   528   534   540
                     546   556   562   568   574   580   590   596   602   608   614   624   630
                     636   642   648   658   664   670   676   682   692   698   704   710   716
                     732   741   750   759   772   781   790   799   809   815   821   827   833
                     839   845   851   857   863   869   882   888   894   900   906   912   918
                     924   930   936   942   963   968   977   982   1000  1015  1025  1035  1045
                    1056  1062  1068  1074  1090  1099  1108  1117  1128  1131  1138  1145  1152
                    1159  1166  1173  1180  1187  1194  1201  1207  1221  1229  1237  1245  1259
                    1264  1274  1279  1291  1295  1310  1320  1328  1332  1347  1362  1366  1373
                    1385  1410  1416  1447  1474  1503  1528  1533  1560  1566  1572  1578  1589
                    1595  1601  1607  1623  1632  1641  1650  1665  1672  1679  1686  1693  1700
                    1707  1714  1721  1728  1741  1749  1757  1765  1780  1786  1796  1828  1847
                    1853  1870  1941  1946  1975  1984  2003  2016  2019  2034  2043  2061  2070
                    2077  2092  2101  2120  2133  2148  2157  2165  2180  2193  2207  2216  2234
                    2243  2251  2294  2296  2301  2310  2312  2313* 2326  2327* 2330  2331* 2333
                    2363  2364* 2415  2416* 2427  2428* 2435  2436* 2438  2466  2467* 2480  2481*
                    2483  2484* 2533  2534* 2539  2540* 2542  2546  2547  2566  2595  2598  2603
                    2609* 2610  2667* 2676
.BIT    = 166000     370*  377   378   379   380   381   382   383   384   385   2261  2527  2531
                    2544
.PR     014776      2563* 2565* 2567* 2568* 2570  2574  2584* 2586  2596* 2597  2600* 2602  2606
                    2609*
.PRF    014716      2572  2587  2592*
.PRL    014674      2585* 2601
.PTIT   014620      2564  2569*
.TYPE   010552      1932* 1933  1959*
```

```
BITYPE      1#    444#
BLKBLK      1#
BUFHDR      1#
C         444#    805    811    817    823    829    835    841    847    853    859    865    878    884    890
          896     902    908    914    920    926    932    938
COMP      444#   1129   1136   1143   1150   1157   1164   1171   1178   1185   1192   1199   1205   1219   1227
         1235    1243   1257   1262   1272   1277   1293   1307   1317   1330   1344   1364   1371   1531   1663
         1670    1677   1684   1691   1698   1705   1712   1719   1726   1739   1747   1755   1763   1778   1784
         1794    2014   2131   2191
C1        444#    725    734    743    752    765    774    783    792   1018   1028   1038   1083   1092   1101
         1110    1616   1625   1634   1643
DECML       1#    444#
DUMP        1#    444#   2331   2436   2537   2548   2554
DUMP18      1#    370#   2376   2477
FILBLK      1#
LNKBLK      1#
ODT11       1#
ODT11X      1#
PFTST     444#   1964   1992   2023   2050   2081   2109   2137   2169   2196   2223
POP         1#    444#   2267   2270
PRINT       1#    444#   2019   2077   2165   2251   2296   2312   2326   2329   2333   2363   2415   2427   2435
         2438    2466   2480   2483   2533   2539
PUSH        1#    444#   2264
REGTST    444#   1557   1563   1569   1575
RESPON    444#   1134   1141   1148   1155   1162   1169   1176   1183   1190   1197   1214   1217   1225   1233
         1241    1252   1255   1267   1270   1286   1298   1323   1335   1350   1393   1423   1431   1450   1455
         1462    1481   1491   1513   1525   1539   1660   1667   1674   1681   1688   1695   1702   1709   1716
         1723    1730   1737   1745   1753   1761   1769   1776   1789   1792   1798   1806   1808   1810   1812
         1816    1826   1829   1840   1857   1868   1871   1881   1890   1898   1901
SCOPE.      1#    514#   548#   582#   616#   650#   684#   721#   761#   801#   874#   947#  1004#  1048#  1076#
         1120#   1209#  1247#  1281#  1387#  1425#  1457#  1486#  1508#  1518#  1552#  1581#  1609#  1653#  1731#
         1770#   1799#  1819#  1832#  1861#  1874#  1882#  1892#  1964#  1992#  2023#  2050#  2081#  2109#  2137#
         2169#   2196#  2223#
SCOP.       1#
SDUMP       1#    444#   2425   2481
SETIT     444#   1216   1224   1232   1240   1254   1269   1744   1752   1760   1791
TIME      444#   1135   1142   1149   1156   1163   1170   1177   1184   1191   1198   1215   1218   1226   1234
         1242    1253   1256   1268   1271   1287   1299   1324   1336   1352   1394   1424   1432   1451   1456
         1463    1483   1492   1514   1526   1540   1661   1668   1675   1682   1689   1696   1703   1710   1717
         1724    1736   1738   1746   1754   1762   1775   1777   1790   1793   1804   1807   1809   1811   1813
         1824    1837   1842   1866   1879   1887   1897   1900
TRACE       1#
TRNBLK      1#
TSTREG    444#    518    552    586    620    654    688   1052   1585
TYPEM       1#
WRITE       1#
$CATCH      1#    370#   444
$CMTAG      1#    370#   502
$DECML      1#
$END        1#
$EQUAT      1#    370#   428
$HLT        1#    370#  2519
$KRAT       1#
$LOADR      1#
$OCTAL      1#    370#  2557
$POWER      1#    370#
```

```
$RAND       1#
$RAND4      1#
$READ       1#
$SCOPE      1#    370#   2253
$SETUP      1#    370#
$SRAT       1#
$SWDOC      1#    370#    374
$TRAP       1#
$TYPE       1#    370#   1919
$URAT       1#
.SCOP       1#
.SCOPE      1#    370#    514     548     582     616     650     684     721     761     801     874     947    1004    1048
         1076    1120   1209    1247    1281    1387    1425    1457    1486    1508    1518    1552    1581    1609    1653
         1731    1770   1799    1819    1832    1861    1874    1882    1892    1964    1992    2023    2050    2081    2109
         2137    2169   2196    2223
```

DZDAAA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADC     1904
ADD      480  1359  1404  1418  1419  1452  1465  1471  1484  1855  1903  1349  1952  2323  2358
         2399  2407  2432  2462
ASL     1397  1398  1435  1436  1469  1470  2368  2373  2374  2470  2475  2476  2576  2577  2578
ASR     2353  2357  2457  2461
ASRB    2579  2581
BCC      493
BEQ      469   497   522   528   534   540   546   556   562   568   574   580   590   596   602
         608   614   624   630   636   642   648   658   664   670   676   682   692   698   704
         710   716   732   741   750   759   772   781   790   799   809   815   821   827   833
         839   845   851   857   863   869   882   888   894   900   906   912   918   924   930
         936   942   963   968   977   982  1000  1015  1025  1035  1045  1056  1062  1068  1074
        1090  1099  1108  1117  1128  1131  1138  1145  1152  1159  1166  1173  1180  1187  1194
        1201  1207  1221  1229  1237  1245  1259  1264  1274  1279  1295  1310  1320  1332  1347
        1354  1366  1373  1378  1385  1410  1416  1447  1474  1478  1503  1533  1560  1566  1572
        1578  1589  1595  1601  1607  1623  1632  1641  1650  1665  1672  1679  1686  1693  1700
        1707  1714  1721  1728  1741  1749  1757  1765  1780  1786  1796  1828  1847  1853  1870
        1909  1935  1984  2016  2043  2070  2101  2133  2157  2193  2216  2243  2275  2289  2310
        2356  2429  2460  2502  2510  2528  2552  2595  2598
BIC      727   730   736   739   745   748   754   757   767   770   776   779   785   788   794
         797   965   979  1013  1020  1023  1030  1033  1040  1043  1085  1088  1094  1097  1103
        1106  1112  1115  1412  1449  1618  1621  1627  1630  1636  1639  1645  1648  1849  1953
        2298  2395  2433
BICB    1370
BIS      967   971
BISB    2583  2599
BIT      468   990   999  1127  1399  1437  1827  1843  1869  1905  1930  2261  2272  2293  2509
        2527  2531  2586
BLT     2417  2419  2485  2487
BNE      467   482   991  1400  1421  1438  1454  1480  1498  1507  1545  1547  1844  1906  1931
        1937  1943  1951  2262  2273  2277  2294  2325  2343  2371  2386  2402  2448  2473  2513
        2515  2532  2587  2601  2603  2686  2688
BPL     1941  1946  2542
BR       484   499  1291  1302  1306  1315  1328  1339  1343  1356  1362  1381  1406  1443  1529
        1856  1947  1975  1986  2003  2012  2034  2045  2061  2072  2092  2103  2120  2129  2148
        2159  2180  2189  2207  2218  2234  2245  2263  2268  2271  2315  2321  2328  2334  2345
        2359  2410  2439  2450  2463  2564  2566  2572  2676
CLC      490
CLR      464   543   577   611   645   679   713   872   873   945   946   952   985   986  1047
        1071  1119  1125  1126  1314  1326  1338  1361  1392  1405  1430  1536  1543  1604  1652
        1658  1659  1805  1825  1839  1867  1880  1889  1991  2108  2354  2365  2366  2380  2459
        2468  2494  2495  2497  2499  2508  2567  2682  2684
CLRB    2571  2585  2605
CMP      466   479   496   521   527   533   539   545   555   561   567   573   579   589   595
         601   607   613   623   629   635   641   647   657   663   669   675   681   691   697
         703   709   715   731   740   749   758   771   780   789   798   808   814   820   826
         832   838   844   650   856   862   868   881   887   893   899   905   911   917   923
         929   935   941   962   967   976   981  1024  1034  1044  1055  1061  1067  1073  1089
        1098  1107  1116  1130  1137  1144  1151  1158  1165  1172  1179  1186  1193  1200  1206
        1220  1228  1236  1244  1258  1263  1273  1278  1292  1294  1305  1309  1316  1319  1329
        1331  1342  1346  1363  1365  1372  1384  1407  1409  1415  1420  1444  1446  1473  1476
        1477  1479  1502  1529  1532  1559  1565  1571  1577  1588  1594  1600  1606  1622  1631
        1640  1649  1664  1671  1678  1685  1692  1699  1706  1713  1720  1727  1740  1748  1756
        1764  1779  1785  1795  1846  1852  1950  1976  1983  1987  2004  2013  2015  2035  2042
        2046  2062  2069  2073  2093  2100  2104  2121  2130  2132  2149  2156  2160  2181  2190
        2192  2208  2215  2219  2235  2242  2246  2269  2316  2322  2336  2342  2347  2411  2416
```

```
        2418  2428  2441  2447  2452  2484  2486  2602
CMPB    1936  2276
COM      719  1082  1615
DEC      481  1497  1506  1544  1546  1942  2324  2370  2385  2401  2472  2514  2685  2687
EMT      429
HALT     445   470   483  1974  1980  2002  2008  2020  2033  2039  2060  2066  2078  2091  2097
        2119  2125  2147  2153  2166  2179  2185  2206  2212  2233  2239  2252  2327  2335  2346
        2420  2440  2451  2488  2518  2543
INC     2309  2389  2512  2530
INCB    2279  2285  2596  2600
IOT      430
JMP      448   449   450   451   452   453   457   459   472   993  1001  1549  1914  2299  2311
        2421  2430  2689
JSR      511   996  1135  1142  1149  1156  1163  1170  1177  1184  1191  1198  1215  1218  1226
        1234  1242  1253  1256  1268  1271  1287  1299  1324  1336  1352  1394  1424  1432  1451
        1456  1463  1483  1492  1514  1516  1526  1540  1661  1668  1675  1682  1689  1696  1703
        1710  1717  1724  1736  1738  1746  1754  1762  1775  1777  1790  1793  1804  1807  1809
        1811  1813  1824  1837  1842  1866  1879  1887  1897  1900  1910  2282  2287  2332  2377
        2426  2437  2478  2482  2538  2540  2550  2555  2673
MOV      461   465   471   474   475   476   488   489   495   510   512   520   525   526   531
         532   537   538   544   554   559   560   565   566   571   572   578   588   593   594
         599   600   605   606   612   622   627   628   633   634   639   640   646   656   661
         662   667   668   673   674   680   690   695   696   701   702   707   708   714   718
         726   728   729   735   737   738   744   746   747   753   755   756   766   768   769
         775   777   778   784   786   787   793   795   796   806   807   812   813   818   819
         824   825   830   831   836   837   842   843   848   849   854   855   860   861   866
         867   879   880   885   886   891   892   897   898   903   904   909   910   915   916
         921   922   927   928   933   934   939   940   953   954   955   956   960   961   966
         974   975   980   995   997  1009  1010  1011  1012  1019  1021  1022  1029  1031  1032
        1039  1041  1042  1053  1054  1059  1060  1065  1066  1072  1081  1084  1086  1087  1093
        1095  1096  1102  1104  1105  1111  1113  1114  1129  1134  1136  1141  1143  1148  1150
        1155  1157  1162  1164  1169  1171  1176  1178  1183  1185  1190  1192  1197  1199  1205
        1214  1216  1217  1219  1224  1225  1227  1232  1233  1235  1240  1241  1243  1252  1254
        1255  1257  1262  1267  1269  1270  1272  1277  1286  1288  1293  1298  1300  1303  1308
        1313  1318  1323  1325  1330  1335  1337  1340  1345  1351  1355  1357  1358  1360  1364
        1369  1371  1379  1380  1383  1393  1395  1401  1402  1403  1408  1414  1422  1423  1431
        1433  1439  1440  1441  1442  1445  1450  1455  1462  1464  1466  1467  1472  1482  1491
        1493  1495  1496  1500  1501  1513  1515  1523  1524  1525  1531  1535  1539  1542  1557
        1558  1563  1564  1569  1570  1575  1576  1580  1586  1587  1592  1593  1598  1599  1605
        1614  1617  1619  1620  1626  1628  1629  1635  1637  1638  1644  1646  1647  1660  1663
        1667  1670  1674  1677  1681  1684  1688  1691  1695  1698  1702  1705  1709  1712  1716
        1719  1723  1726  1730  1737  1739  1744  1745  1747  1752  1753  1755  1760  1761  1763
        1768  1769  1776  1778  1783  1784  1789  1791  1792  1794  1798  1806  1808  1810  1812
        1814  1815  1816  1826  1829  1839  1841  1845  1850  1851  1858  1868  1871  1881  1990
        1898  1901  1908  1927  1928  1929  1932  1933  1948  1954  1955  1956  1961  1962  1969
        1970  1971  1972  1977  1978  1979  1981  1982  1988  1990  1997  1998  1999  2000  2005
        2006  2007  2009  2010  2014  2018  2021  2028  2029  2030  2031  2036  2037  2038  2040
        2041  2047  2049  2055  2056  2057  2058  2063  2064  2065  2067  2068  2074  2076  2079
        2086  2087  2088  2039  2094  2095  2096  2098  2099  2105  2107  2114  2115  2116  2117
        2122  2123  2124  2126  2127  2131  2135  2136  2142  2143  2144  2145  2150  2151  2152
        2154  2155  2161  2163  2167  2174  2175  2176  2177  2182  2183  2184  2186  2187  2191
        2195  2201  2202  2203  2204  2209  2210  2211  2213  2214  2220  2222  2228  2229  2230
        2231  2236  2237  2238  2240  2241  2247  2249  2264  2265  2267  2270  2280  2281  2286
        2290  2302  2303  2304  2305  2306  2307  2308  2313  2317  2318  2319  2331  2337  2338
        2339  2341  2344  2348  2349  2350  2360  2361  2364  2367  2372  2375  2376  2378  2381
        2382  2383  2384  2387  2388  2390  2396  2397  2398  2400  2403  2404  2405  2406  2408
```

```
        2409  2412  2413  2414  2422  2424  2425  2431  2434  2436  2442  2443  2444  2446  2449
        2453  2454  2464  2467  2469  2474  2477  2479  2481  2489  2491  2492  2493  2496  2438
        2500  2503  2507  2516  2534  2537  2549  2554  2563  2569  2570  2573  2574  2584  2607
        2670  2671  2672  2674  2675  2683
MOVB    1002  1204  1550  1938  1939  1944  2278  2297  2536  2565  2568  2575  2604
NEG     1396  1434  1468  2351  2455
NOP      471   958   972   998  1911  1912  1913  2677  2678  2679  2680  2681
RESET    462   519   553   587   621   655   689   994  2669
ROL      491   492  2369  2471  2588  2590  2592
ROLB    2589  2591  2593
ROR     2391  2392  2393  2394  2580  2582
RTI     1957  2283  2291  2544
RTS      478   494  2295  2504  2511  2556  2608
SUB      463  1382  1413  1453  1499  2535
SWAB    2352  2456
TRAP     428
TST      477  1014  1289  1301  1353  1377  1505  1530  2266  2288  2314  2320  2340  2355  2373
        2423  2445  2459  2490  2501  2541  2551
TSTB    1934  1940  1945  2274  2594  2597
WAIT    1973  2001  2032  2059  2090  2118  2146  2178  2205  2232
.ASCII  2636
.ASCIZ  2020  2078  2166  2252  2297  2313  2327  2331  2334  2364  2416  2428  2436  2439  2467
        2481  2484  2534  2540
.BLKW   2609
.ENABL     1   370
.END    2692
.ENDC    444   525   559   593   627   661   695  1053  1586  1667  1674  1681  1688  1695  1702
        1709  1716  1934  1958  1964  1968  1986  1990  1996  2009  2018  2023  2027  2045  2049
        2050  2054  2072  2076  2081  2085  2103  2107  2113  2126  2135  2137  2141  2159  2163
        2173  2186  2195  2196  2200  2218  2222  2223  2227  2245  2249  2261  2262  2300  2531
        2533  2541  2545  2557  2585  2611
.EVEN   2020  2078  2166  2252  2297  2313  2327  2331  2334  2364  2416  2428  2436  2439  2467
        2481  2484  2534  2540
.IF      444   519   553   587   621   655   689  1053  1586  1663  1670  1677  1684  1691  1698
        1705  1712  1930  1948  1964  1981  1992  1996  2009  2023  2040  2050  2067  2081  2098
        2109  2113  2126  2137  2154  2169  2173  2186  2196  2213  2223  2240  2261  2292  2527
        2531  2540  2544  2557  2573  2610
.IFF    1667  1670  1677  1684  1691  1698  1705  1712  1958  1964  1986  1987  1988  1992  1996
        2009  2014  2023  2045  2046  2047  2050  2054  2067  2073  2074  2081  2085  2098  2104
        2105  2109  2113  2126  2131  2137  2159  2160  2161  2169  2173  2186  2191  2196  2200
        2213  2219  2220  2223  2245  2246  2247  2261  2527  2545  2557
.IFT    1988  2018  2047  2074  2105  2135  2161  2195  2220  2247
.IFTF   1987  2013  2046  2073  2104  2130  2160  2190  2219  2246
.IIF     377   378   379   380   381   382   383   384   385   811   817   823   829   835   841
         847   884   890   896   902   908   914   920  1134  1141  1148  1155  1162  1169  1176
        1183  1663  1670  1677  1684  1691  1698  1705  1712  2282  2287  2301  2546  2547  2610
.IRP    2264  2267  2270
.LIST      1   370   444   445   501   514   518   548   552   582   586   616   620   650   654
         684   688   721   725   761   765   801   805   811   817   823   829   835   841   847
         853   874   878   884   890   896   902   908   914   920   926   947   951   994  1004
        1008  1048  1052  1076  1080  1120  1124  1134  1136  1141  1143  1148  1150  1155  1157
        1162  1164  1169  1171  1176  1178  1183  1185  1190  1192  1199  1209  1213  1216  1219
        1227  1235  1243  1247  1251  1254  1257  1269  1272  1281  1285  1288  1300  1325  1337
        1383  1387  1391  1395  1425  1429  1433  1452  1457  1461  1464  1484  1486  1490  1493
        1508  1512  1515  1518  1522  1527  1541  1550  1552  1556  1581  1585  1609  1613  1653
        1657  1662  1663  1669  1670  1676  1677  1683  1684  1690  1691  1697  1698  1704  1705
```

```
        1711  1712  1718  1719  1725  1731  1735  1737  1739  1747  1755  1763  1770  1774  1776
        1778  1791  1794  1799  1803  1805  1808  1810  1812  1814  1819  1823  1925  1832  1836
        1838  1843  1861  1865  1867  1874  1878  1880  1882  1886  1888  1892  1896  1898  1901
        1919  1961  1964  1968  1992  1996  2020  2021  2023  2027  2050  2054  2078  2079  2081
        2085  2109  2113  2137  2141  2166  2167  2169  2173  2196  2200  2223  2227  2252  2253
        2297  2302  2313  2327  2331  2334  2364  2416  2428  2436  2439  2467  2481  2484  2519
        2534  2540  2557  2611
.MACRO     1   444
.MCALL   370   444   514   548   582   616   650   684   721   761   801   874   947  1004  1048
        1076  1120  1209  1247  1281  1387  1425  1457  1486  1508  1518  1552  1581  1609  1653
        1731  1770  1799  1819  1832  1861  1874  1882  1892  1964  1992  2023  2050  2081  2109
        2137  2169  2196  2223
.NLIST     1   370   444   445   501   514   518   548   552   582   586   616   620   650   654
         684   689   721   725   761   765   801   805   811   817   823   829   835   841   847
         853   874   878   884   890   896   902   908   914   920   926   947   951   994  1004
        1008  1048  1052  1076  1080  1120  1124  1134  1136  1141  1143  1148  1150  1155  1157
        1162  1164  1169  1171  1176  1178  1183  1185  1190  1192  1199  1209  1213  1216  1219
        1227  1235  1243  1247  1251  1254  1257  1269  1272  1281  1285  1288  1300  1325  1337
        1353  1387  1391  1395  1425  1429  1433  1452  1457  1461  1464  1484  1486  1490  1493
        1508  1512  1515  1518  1522  1527  1541  1550  1552  1556  1581  1585  1609  1613  1653
        1657  1662  1663  1669  1670  1676  1677  1683  1684  1690  1691  1697  1698  1704  1705
        1711  1712  1718  1719  1725  1731  1735  1737  1739  1747  1755  1763  1770  1774  1776
        1778  1791  1794  1799  1803  1805  1808  1810  1812  1814  1819  1823  1825  1832  1836
        1838  1843  1861  1865  1867  1874  1878  1880  1882  1886  1888  1892  1896  1898  1901
        1919  1961  1964  1968  1992  1996  2020  2021  2023  2027  2050  2054  2078  2079  2081
        2085  2109  2113  2137  2141  2166  2167  2169  2173  2196  2200  2223  2227  2252  2253
        2297  2302  2313  2327  2331  2334  2364  2416  2428  2436  2439  2467  2481  2484  2519
        2534  2540  2557  2611
.PAGE    408   444   548   582   616   650   684   718   761   801   841   874   914   947  1048
        1076  1120  1169  1209  1247  1281  1387  1425  1508  1581  1609  1653  1698  1731  1770
        1799  1882  1902  1992  2021  2050  2079  2109  2137  2167  2196  2223  2302  2557  2611
.REM       1
.REPT    445   811   884  1134  1663
.SBTTL   444   501   514   548   582   616   650   684   721   761   801   874   947   994  1004
        1048  1076  1120  1209  1247  1281  1387  1425  1457  1486  1508  1518  1550  1552  1581
        1609  1653  1731  1770  1799  1819  1832  1861  1874  1882  1892  1919  1964  1992  2023
        2050  2081  2109  2137  2169  2196  2223  2253  2302  2519  2557  2611
.TITLE   370
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
*,DZDAAA.SEQ/SOL/CRF/PAGNUM/NL:TOC=STEVE.SML,DZDAAA.P11
RUN-TIME: 27 37 5 SECONDS
RUN-TIME RATIO: 379/70=5.3
CORE USED:  27K  (53 PAGES)
```

Spooler runtime 11 Seconds, 48 KCS, 303 disk reads, 4 disk writes, 82 pages