DR11W

DR11-W GEN NPR INTFC

CZDRLA0

AH-E780A-MC

FICHE 1 OF 1

JAN 1980

digital

MADE IN USA

```
                .REM %
```

IDENTIFICATION

PRODUCT CODE:	AC-E779A-MC
PRODUCT NAME:	CZDRLA0 DR11-W GEN NPR INTFC
DATE RELEASED:	AUG. 1979
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	JOHN W. CIUKAJ

TABLE OF CONTENTS

```
                        1.0     ABSTRACT
                                ---------

                                THIS DIAGNOSTIC PROGRAM IS CAPABLE OF TESTING EITHER THE DR11W
                                OR THE DR11B NPR GENERAL INTERFACE.

                                IT HAS THE FOLLOWING FEATURES:

                                1.  ACT11/XXDP COMPATIBLE
                                2.  MULTIPLE BOARD TESTING USING TABLE CREATED BY USER
                                3.  BURST DATA LATE CALIBRATION
                                4.  INDEPENDANT 'LOGIC WRAP-AROUND' AND
                                    'CABLE WRAP-AROUND' TESTING


                        2.0     REQUIREMENTS
                                ------------

                        2.1     [EQUIPMENT]

                                1.  PDP11 STANDARD COMPUTER
                                2.  I/O TYPE TERMINAL
                                3.  DR11-W
                                4.  LOOP BACK CABLE (OPTIONAL)

                        2.2     [STORAGE]

                                THE PROGRAM USES 4600 WORDS OF MEMORY

                        3.0     TESTING MODES
                                -------------

                        3.1     [DEFINITION]

                                THE DR11W DIAGNOSTIC PROVIDES TWO INDEPENDANT MODES OF LOGIC TESTING:

                                1.  MAINTENANCE MODE TESTING
                                2.  CABLE MODE TESTING

                                IN CONTEXT OF THE DIAGNOSTIC,'MAINTENANCE MODE TESTING'
                                IS DEFINED AS TESTS WHICH ARE PERFORMED WITH BIT 12 OF THE CSR SET.
                                THIS RESULTS IN 'LOGIC WRAP-AROUND' WHICH ALLOWS TESTING
                                WITHOUT A USER DEVICE OR PHYSICAL CABLE.  LOGIC WRAP-AROUND
                                CHECKS ALL LOGIC FUNCTIONALITY EXCEPT MODULE CONNECTORS,CABLES,
                                AND I/O DRIVERS AND RECEIVERS.  'CABLE MODE TESTING' ,THEREFORE,CHECKS
                                ALL THESE THAT THE LOGIC WRAP-AROUND DOESN'T.

                        3.2     [IMPLEMENTATION]

                                THE TWO MODES OF TESTING ARE INVOKED AS FOLLOWS:

                                1.  THE PROGRAM WILL ALWAYS DO THE MAINTENANCE MODE
                                                    ------
                                    TESTS(LOGIC WRAP-AROUND)
```

2. IF SWITCH REGISTERS 9 AND 10 ARE 0, THEN THE PROGRAM WILL AUTOMATICALLY CHECK IF THE PHYSICAL WRAP-AROUND CABLE IS IN PLACE. IF IT IS, 'CABLE TESTS ' WILL BE DONE. IF NOT,CABLE TESTS ARE SKIPPED.

NOTE: STEP 2 CAN BE ALTERED BY USING SWITCH REGISTERS 9 OR 10. SEE SECT. 5.1

4.0 LOAD AND START PROCEDURE

1. LOAD PROGRAM INTO MEMORY
2. LOAD STARTING ADDRESS 200
3. PRESS START-SEE SECT 7.0 FOR OPERATING MODES

5.0 SOFTWARE SWITCH REGISTER

5.1 [OPTIONS]

SWITCH	OCTAL	FUNCTION
SW15=1	100000	HALT ON ERROR
SW14=1	040000	LOOP ON TEST
SW13=1	020000	INHIBIT ERROR TYPEOUTS
SW11=1	004000	INHIBIT ITERATIONS
SW10=1	002000	PERFORM MAINT. MODE (INTERNAL LOGIC WRAP-AROUND) TESTS ONLY.
SW09=1	001000	PERFORM CABLE MODE TESTS UNCONDITIONALLY (DO NOT ALLOW PROGRAM TO DECIDE WHETHER CABLE IS IN)
SW08-1	000400	AFTER 1ST OCCURRENCE OF ERROR DURING A PASS OF PROGRAM,JUMP TO THE NEXT BOARD SPECIFIED IN TABLE FOR TESTING.

5.2 [IMPLEMENTATION]

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

5.3 [CONTROL]

THIS PROGRAM SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC 176) FROM THE TTY. THIS IS ACCOMPLISHED AS FOLOWS:

1. TYPE CONTROL G <^G>;THIS ALLOWS THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2. THE MACHINE WILL TYPE: SWR=XXXXXX NEW (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTW. SWITCH REGISTER)

3. AFTER THE 'NEW ' THE OPERATOR CAN DO ONE OF THE FOLLOWING:

A. TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>
(ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
WILL BE ALLOWED).
IF A <CR> IS THE FIRST ENTRY THE SOFTWARE SWITCH REGISTER WILL
NOT BE CHANGED.

B. IF A CONTROL U <^U> IS DEPRESSED THE PROGRAM WILL GO BACK
TO STEP 2.

5.4 [PROGRAM AND/OR OPERATOR ACTION]

LOADING AND STARTING AT 200 WITH ALL SWITCHES DOWN
IS NORMAL LOGIC TESTING. IF AN ERROR
IS DETECTED HERE, THERE WILL BE A PRINTOUT. WHEN
AN ERROR IS DETECTED AND IT IS NECESSARY TO SCOPE
ON IT, PLACE SW15 UP TO HALT ON ERROR, THEN SW14 UP
TO LOOP ON ERROR, THEN SW13 UP TO DELETE PRINTOUTS.

6.0 ERROR REPORTING

DEPENDING ON WHICH BOARD IS BEING TESTED,THE STATUS PRINTOUT WILL
REFLECT EITHER THE DR11W OR THE DR11B.

ALL ERRORS ARE ACCOMPANIED WITH THE FOLLOWING PRINTOUT:

TESTNO	ERRPC	PSW	DR11W/B STATUS

WHERE:

TESTNO -TEST NUMBER WHERE ERROR OCCURRED
ERRPC - LISTING ADDRESS WHERE THE ERROR WAS DETECTED
PSW -PROCESSOR STATUS WORD AT ERROR TIME
DR11W/B STATUS -CONTENTS OF DR11W/B CONTROL STATUS
REGISTER AT ERROR TIME

7.0 OPERATING MODES

7.1 [MANUAL MODE]

DEFINED AS NON-AUTOMATIC USE OF DIAGNOSTIC.FOR EXAMPLE:

A.LOADING PROGRAM VIA PAPER TAPE
B.LOADING FROM XXDP PACKAGE VIA KEYBOARD
C.REQUESTING A PROGRAM FROM ACT11 VIA ACT11 STATION SWITCHES

ONLY UNDER MANUAL MODE DOES THE DIAGNOSTIC OFFER 'BURST
DATA LATE' CALIBRATION AND 'MULTIPLE BOARD ' DIALOGUE.
AFTER LOADING PROGRAM, DEPOSITING SA 200, AND PRESSING
START, THE PROGRAM TYPES THE FOLLOWING(NOTE:USER INPUT
IS UNDERSCORED):

CZDRL DR11-W GEN NPR INTFC

```
          DO YOU WISH BURST DATA LATE CALIBRATION?
          (Y=YES,N=NO):   N
                          -

          DO YOU WISH MULTIPLE BOARD DIALOGUE?
          (Y-YES,N=NO):   N
                          -


        IF SOFTWARE SWITCH REGISTER IS INVOKED, THEN THE FOLLOWING WILL BE
        PRINTED:

          SWR XXXXXX   NEW=

          (REFER TO SECT. 5.0 FOR OPERATOR OPTIONS)

        ANSWERING 'NO' TO THE ABOVE QUESTIONS(AND ASSUMING THAT
        MULTIPLE BOARD SIZING HAD NOT BEEN PREVIOUSLY DONE) THE
        PROGRAM DEFAULTS TO ONE BOARD TESTING SPECIFIED BY REGISTER
        ADDRESS 'DEFRAD'(172410) AND VECTOR ADDRESS 'DEFVAD'(124).  THE
        PROGRAM THEN TYPES:

          *BRD ADDR.: 172410


        IF NO SWITCH REGISTER OPTIONS ARE SELECTED THE PROGRAM
        WILL 'END PASS' THE FIRST TIME WITH NO ITERATIONS.  SUBSEQUENT
        'END PASS' WILL BE WITH ITERATIONS.


        THE 'END PASS' TYPEOUT IS AS FOLLOWS(IN THE CASE OF  MAINT MODE TESTING):

          TEST MODE: MAINT MODE ONLY
           END PASS # XXXXXX



7.1.1   [MANUAL MODE W/MULTIPLE BOARD FEATURE]

        THE DIAGNOSTIC CONTAINS AN INTERACTIVE MONITOR WHICH
        ALLOWS THE USER TO CREATE AND MODIFY A TABLE DESCRIBING
        UP TO 16 BOARD REGISTER AND VECTOR ADDRESSES FROM WHICH THE PROGRAM
        WILL SUBSEQUENTLY RUN.  THE PROGRAM WILL SEQUENTIALLY RUN THRU EACH BOARD
        DESCRIBED IN THE TABLE FOR 'ENDPAS #1' AND THEN DO THE SAME FOR 'ENDPAS #2',ETC.
        ALL BOARDS FOR 'ENDPAS #1' EXPERIENCE TESTING WITH NO ITERATIONS.  THUS,
        TYPING 'Y'' TO THE QUESTION 'DO YOU WISH MULTIPLE BOARD DIALOGUE?'' ENTERS
        THE USER INTO THE MULTIPLE BOARD MONITOR WITH THE PROMPT '' '' TO WHICH THE
        FOLLOWING COMMANDS MAY BE TYPED:



          ' L ' - LIST PRESENT MULTIPLE BOARD TABLE
          ' E ' - EDIT PRESENT MULT BRD TBL
          ' R ' - START PROGRAM TESTING

        AT ANY POINT DURING PROGRAM RUN A 'RUN SUMMARY' MAY BE RECEIVED
        BY TYPING A 'CONTROL S' <^S>. THE PROGRAM WILL DETECT
```

```
                                THE ^S AND PRINT:

                                  RUN SUMMARY.....

                                  BOARD  PASCNT  ERRCNT
                                  XXXXXX  XXX    XXX
                                  XXXXXX  XXX    XXX
                                  ETC.    ETC.    ETC.

                                WHERE:

                                  BOARD         - LISTING OF ALL BOARD  REGISTER ADDRESSES SPECIFIED
                                                  IN MULTIPLE BOARD TABLE
                                  PASCNT        - # PASSES DONE ON EACH BOARD SPECIFIED
                                  ERRCNT        - # OF PASCNT'S WHERE ONE OR MORE ERRORS WERE DETECTED

                        7.1.2   [MANUAL MODE W/BURST DATA LATE CALIBR.]

                                THE DIAGNOSTIC CONTAINS A ROUTINE WHICH AIDS THE USER
                                IN 'BURST DATA LATE' CALIBRATION.  AS STATED IN THE DR11-W
                                ENGINEERING SPECIFICATION, THE 'BURST DLT' MULTIVIBRATOR
                                TIME OUT MUST BE CALIBRATED SO AS TO BE COMPATIBLE WITH
                                THE USER DEFINED TRANSFER RATE IN BURST MODE OPERATION.
                                EFFECTIVELY, THE PROGRAM SOFTWARE ROUTINE SETS THE CYCLE BIT IN THE CSR
                                OF TH DR11W, FOLLOWS WITH A DELAY, AND THEN CLEARS THE CYCLE BIT.
                                THIS CONTINUES REPEATEDLY UNTIL THE USER HALTS THE CPU.  THUS, WHEN THE
                                USER SUBSEQUENTLY ATTACHES A SCOPE PROBE TO E88-6 ON THE DR11-W (REFER
                                TO PRINT SET M8716-0-1) A POSITIVE PULSE WILL BE OBSERVED.  THE PULSE
                                SHOULD BE SET BETWEEN 5-30 US. BY ADJUSTING POT.  R88.

                                THEREFORE, ANSWERING 'Y' TO THE QUESTION: 'DO YOU WISH
                                BURST DATA LATE CALIBRATION?' WILL RESULT IN THE FOLLOWING PRINTOUT:

                                  BURST DATA LATE CALIBRATION IN PROGRESS..
                                  ATTACH SCOPE PROBE...
                                  TO ABORT THIS ROUTINE, HALT PROCESSOR...


                                THE CALIBRATION ROUTINE WILL FUNCTION ON ONE BOARD SPECIFIED
                                BY 'DEFRAD'(172410) AND 'DEFVAD'(124).



                        7.2     [AUTO MODE]

                                DEFINED AS USING THE PROGRAM IN A SITUATION OF SEQUENTIAL
                                EXECUTION OF TWO OR MORE PROGRAMS ACCORDING TO A PREDETERMINED SEQUENCE
                                AND WITHOUT OPERATOR INTERVENTION.

                                AUTOMATIC MODES ARE:

                                A.  ACT11 QUICK VERIFY
                                B.  ACT11 AUTO ACCEPT
                                C.  XXDP CHAIN MODE
```

THE DR11W DIAGNOSTIC HAS THE FOLLOWING RUN CHARACTERISTICS WHEN OPERATING IN AUTO MODE:

A. THE PROGRAM WILL TEST ONLY ONE BOARD SPECIFIED BY 'DEFRAD' AND 'DEFVAD'.

B. THE FOLLOWING WILL NOT BE OFFERED TO THE USER:

1. BURST DATA LATE CALIBRATION
2. MULTIPLE BOARD DIALOGUE
3. RUN SUMMARY <^S>

8.0 MISCELLANEOUS

8.1 [RESTARTING PROGRAM IN CORE]

WHENEVER THE PROGRAM IS HALTED AND RESTARTED AT SA 200, ALL HISTORY OF PREVIOUS TESTING IS LOST. HOWEVER, THE MULTIPLE BOARD TABLE AS PREVIOUSLY EDITED IS LEFT INTACT. IT WILL REMAIN INTACT UNTIL:

1. ANOTHER PROGRAM IS LOADED INTO CORE
2. THE USER RE-EDITS THE TABLE

8.2 [POWER FAIL]

THERE IS NO RESTART MESSAGE NOR PROVISIONS FOR AUTOMATICALLY RESTARTING PROGRAM UPON POWER UP. IF NON VOLATILE MEMORY, THEN RESTART AT SA 200.

9.0 EXECUTION TIMES

ON A PDP11/04:

1. MAINT MODE ONLY/NO ITER.: APPROX. 5 SEC.
2. MAINT MODE ONLY/WITH ITER.: APPROX. 35 SEC.
3. MAINT & CABLE MODE/NO ITER.: APPROX. 7 SEC.
4. MAINT & CABLE MODE/WITH ITER.: APPROX. 60 SEC..

10.0. SUBROUTINE ABSTRACTS

10.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS RE-QUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE

WILL BE EITHER A FIXED OR RANDOM NUMBER OF ITERATIONS
ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED.
SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS.
 NOTE: SUPPORTS ^G ROUTINE FOR DYNAMIC LOADING
 OF SOFTWARE SWITCH REGISTER

10.2 HALT

IS A ROUTINE THAT PRINTS-OUT AN ADDRESS THAT TAGS
THE FAILING SUBTEST,THE CP STATUS REGISTER AND
THE DR11W STATUS REGISTER AT THE TIME OF FAILURE.
 NOTE: SUPPORTS ^G ROUTINE FOR DYNAMIC LOADING
 OF SOFTWARE SWITCH REGISTER

10.3 LODBUF

THE INBUF BUFFER IS LOADED WITH AN INCREMENTING
PATTERN (0,1,2,3,...) BEGINNING AT THE STARTING
ADDRESS OF INBUF. THE NUMBER OF WORDS LOADED IS
DETERMINED BY THE CONTENTS OF BUFLEN.

10.4 CHKBFF

THE CHKBUF BUFFER IS LOADED WITH A MODIFIED
INCREMENTING PATTERN (0,0,2,2,4,4,6,6,...)
BEGINNING AT THE STARTING ADDRESS OF CHKBUF.
THE NUMBER OF WORDS LOADED IS DETERMINED BY THE
CONTENTS OF BUFLEN. THIS BUFFER IS LOADED ONLY
FOR TESTS WHICH USE THE MAINTENANCE MODE OF THE
DR11-W WHICH HAS A SPECIAL ALTERNATING DATI-DATO
 SEQUENCE OF OPERATION.

10.5 INTA

THE IE BIT IS CLEARED IN THE CSR THEN THE CSR
IS CHECKED FOR THE ABSENCE OF AN ERROR AND THE
PRESENCE OF READY. THE WCR IS CHECKED TO SEE
THAT IT IS EQUAL TO ZERO. THE CORRECT CONTENTS
OF THE BAR ARE CALCULATED AND CHECKED. THERE
IS A JSR TO THE NORMAL SUB-ROUTINE BEFORE THIS
ROUTINE IS EXITED. THE PROGRAM WILL HALT IF
ERROR IS SET, READY IS CLEAR, OR READY AND ERROR
ARE CLEAR.

10.6 DATCHK

THIS ROUTINE IS ENTERED TO CHECK INBUF AFTER
A MAINTENANCE MODE OPERATION. THE CONTENTS OF
INBUF AND THE CONTENTS OF CHKBUF ARE CHECKED TO
SEE THAT THEY ARE THE SAME. THE NUMBER OF COM-
PARISONS MADE IS DETERMINED BY THE CONTENTS OF
BUFLEN.

10.7 NORMAL

THE ROUTINE IS ENTERED FROM INTA AND FROM SOME
TESTS WHICH DON'T USE INTA. THE NUMBER OF

THE DRINV+2 IS PUT INTO DRINV AND THE DRVS IS
CLEARED. IF THE DR11-W INTERRUPTS UNDER THESE
CONDITIONS THE PDP-11 WILL HALT AT DRVS. THE
PROCESSOR STATUS WORD IS RESTORED TO LEVEL 7 AND
THE ROUTINE IS EXITED.

10.8 DATOCK

AFTER A STRING OF DATO'S HAS BEEN COMPLETED THIS
ROUTINE CHECKS THAT THE CORRECT DATA PATTEPN
(52525) WAS TRANSFERRED TO INBUF. THE NUMBER
OF COMPARISONS MADE IS DETERMINED BY THE CONTENTS
OF BUFLEN. AN ADDITIONAL CHECK IS MADE ON BUFLEN+2
TO INSURE THAT TOO MANY WORDS WEREN'T TRANSFERRED.

10.9 ERRCHK

THIS ROUTINE CLEARS IE AND HALTS IF ERROR IS SET.

10.10 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0
DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND
INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF
MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS
OF THE NEXT LOCATION. THE NEXT LOCATION IS LOADED WITH
A HALT (000000). THUS AN ILLEGAL TRAP OR INTERRUPT WILL
CAUSE A HALT AT THE TRAP LOCATION PLUS TWO.

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA, EXAMINE
REGISTER SIX, IT WILL CONTAIN THE CURRENT STACK ADDRESS.
THE CONTENTS OF THE CURRENT STACK ADDRESS IS THE VALUE OF
THE LOCATION COUNTER WHEN THE TRAP OR INTERRUPT OCCURRED.

11.0 EXAMPLES

11.1 [USING MULTIPLE BOARD FEATURE (REFER TO SECT. 7.1.1)]

(USER LOADS PROGRAM AND STARTS AT 200)

```
CZDRL   DR11-W GEN NPR INTFC

DO YOU WISH BURST DATA LATE CALIBRATION?
(Y-YES,  N=NO):  N
                 -

DO YOU WISH MULTIPLE BOARD DIALOGUE?
(Y-YES,  N NO):  Y
                 -


    L
    -
```

```
548                                     NO. BOARDS REQUESTED FOR TESTING:  1
549
550                                     BOARD#   REGSTR.   VECTOR
551
552                                          1   172410    000124
553
554                                     - E
555                                       -
556
557                                     HOW MANY BOARDS DO YO WANT TO TEST?   3 <CR>
558                                                                           -
559
560                                     BOARD 1
561                                     REGISTER ADDRESS:  172410     <CR>  <---(USER INPUTS <CR>:CONTENTS STAY SAME)
562                                     VECTOR ADDRESS:       000124  <CR>
563
564                                     BOARD  2
565                                     REGISTER ADDRESS:  000000     172430 <CR>  <---( 0 CONTENTS ARE CHANGED)
566                                     VECTOR ADDRESS:       000000  170 <CR>
567
568                                     BOARD  3
569                                     REGISTER ADDRESS:  000000     172450 <CR>
570                                     VECTOR ADDRESS:       000000  174 <CR>
571
572                                     - L
573                                       -
574
575                                     BOARD #   REGSTR.  VECTOR
576
577                                           1   172410   000124
578                                           2   172430   000170
579                                           3   172450   000174
580
581                                       R
582                                       -
583
584                                         (PROGRAM NOW RUNS USING MULT. BOARD TABLE)
585
586
587                                     *BRD ADDR.:   172410
588
589                                     TEST MODE:    MAINT & CABLE
590                                     END PASS      #     1
591
592                                     *BRD ADDR.:   172430
593
594                                     TEST MODE:    MAINT & CABLE
595                                     END PASS      #     1
596
597                                     *BRD ADDR.:   172450
598
599                                     TEST MODE:    MAINT & CABLE
600                                     END PASS      #     1
601
602                                     *BRD ADDR.:   172410
603
604                                     TEST MODE:    MAINT & CABLE
```

```
                        END PASS      #     2

                        *BRD ADDR.:   172430

                        TEST MODE:    MAINT & CABLE
                        END PASS      #     2

                        *BRD ADDR.:   172450

                        TEST MODE:    MAINT & CABLE
                        END PASS      #     2

                              .
                              .
                              .
                              ETC.
                              .

                              (AT ANY TIME USER MAY REQUEST 'RUN SUMMARY' BY TYPING ^S)

                              .
                        ^S

                        RUN SUMMARY.....

                         BOARD    PASCNT    ERRCNT
                       172410     10          0
                       172430     10          0
                       172450      9          0


                              (AFTER RUN SUMMARY,PROGRAM RESTARTS
                               AUTOMATICALLY AT SA 200)

                        CZDRL    DR11-W GEN NPR INTFC

                        DO YOU WISH BURST DATA LATE CALIBRATION?
                        (Y=YES,  N=NO):  N
                                         -
                                    .
                                    .
                                    .
                                    ETC.
```

```
    649                             12.0    PROGRAM SUMMARY
    650                                     ---------------
    651
    652
    653
    654                             %
```

```
                                .TITLE  CZDRLA0-DR11W GEN NPR INTFC
                                ;*COPYRIGHT (C) 1979
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY JOHN W. CIUKAJ
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
                                $TN 1
000001                          $SWR 160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
160000
000240                                  NOP=000240
000004                                  SCOPE=IOT
104000                                  HLT=EMT
000001                                  BIT0=000001
000002                                  BIT1=000002
000004                                  BIT2=000004
000010                                  BIT3=000010
000020                                  BIT4=000020
000040                                  BIT5=000040
000100                                  BIT6=000100
000200                                  BIT7=000200
000400                                  BIT8=000400
001000                                  BIT9=001000
002000                                  BIT10=002000
004000                                  BIT11=004000
010000                                  BIT12=010000
020000                                  BIT13=020000
040000                                  BIT14=040000
100000                                  BIT15=100000

000004                                  BUSERR=000004
000000                  R0=%0
000001                  R1=%1
000002                  R2=%2
000003                  R3=%3
000004                  R4=%4
000005                  R5=%5
000006                  R6=%6
000007                  R7 %7
000007                  PC %7
000006                  SP %6

000001                  GO=1
000002                  FNCT1=2
000004                  FNCT2 4
000010                  FNCT3=10
000020                  XBA16=20
000040                  XBA17=40
000100                  IE=100
000200                  READY=200
000400                  CYCLE=400
001000                  DSTATA=1000
002000                  DSTATB=2000
004000                  DSTATC=4000
010000                  MAINT=10000
```

```
                    020000                  ATTN=20000
                    040000                  NEX=40000
                    100000                  ERROR=100000
                    000200                      CRLF=200
                    022404                  STACK=BUFF
                    000034                  TRAPVEC=34
                    000004                  ERRVEC=4
                    177570                  DSWR=177570
                    177570                  DDISP=177570
                    000060                  TKVEC=60

                                            ;LOAD TRAP CATCHER INTO 0 THRU 777.

000000                                          .ASECT
                    000000                      .=0
                    000020                      .=20
000020  014030                                  SCOPEC
000022  000340                                  340
                    000030                      .-30
000030  013314                                  PRINT
000032  000340                                  340
                    000034                      .-34
000034  017176                                  $TRAP
000036  000340                                  340
                    000046                      .=46
000046  013250                                  $ENDAD
                    000052                      .=52
000052  000000                                  0
                    000174                      .=174
000174  000000                                  DISPREG:        0
000176  000000                                  SWREG:  0
                    000200                      . 200
000200  012706  022404                          MOV     #STACK,SP
000204  005077  000572                          CLR     @PSW
000210  005737  000042                          TST     @#42            ;ARE WE IN AUTO MODE
000214  001411                                  BEQ     1$              ;IF NOT,DO TITLE,CALIBRATION
                                                                        ;AND MULTIPLE BOARD SIZING
000216  023737  000042  000046                  CMP     @#42,@#46       ;IS THIS ACT11 AUTO MODE
000224  001415                                  BEQ     2$              ;IF SO ,DON'T PRINT TITLE
000226  012702  014646                          MOV     #$TITLE,R2
000232  004737  015372                          JSR     PC,TTOUT
000236  000410                                  BR      2$
000240  012702  014646          1$:             MOV     #$TITLE,R2      ;PRINT THE TITLE

000244  004737  015372                          JSR     PC,TTOUT
000250  004737  020540                          JSR     PC,BRSTDL       ;BURST DATA LATE CALIBRATION?
000254  004737  017256                          JSR     PC,MBD          ;MULTIPLE BOARD DIALOGUE
000260  005037  001200          2$:             CLR     FLAG
000264  000137  001364                          JMP     SCSWR
                    001000                      .=1000
001000  177570                  SR:             177570
001002  177776                  PSW:            177776
001004  172410                  DEFRAD: 172410          ;DEFAULT REGISTER ADDRESS
001006  000124                  DEFVAD: 124             ;DEFAULT VECTOR ADDRESS
001010  000000                  PNTRAD: 0               ;POINTER INTO MULTIPLE BOARD TABLE SPECIFYING
                                                        ;CURRENT REGISTER ADDRESS BEING TESTED
001012  000000                  PNTVAD: 0               ;POINTER FOR VECTOR ADDRESS
```

```
 781 001014  000000                PNTPAS: 0               ;POINTER INTO PASCNT TABLE
 782 001016  000000                PNTERR: 0               ;POINTER INTO ERROR COUNT TABLE
 783 001020  000000                PRGCYC: 0               ;COUNTER TO TRACK WHEN ALL BOARDS ARE TESTED
 784 001022  000001                QTYBRD: 1               ;TOTAL # DR11W BOARDS BEING TESTED
 785 001024  172410                REGADR: 172410
 786 001026                              .BLKW   15.       ;TOTAL:16 LOCATIONS FOR BOARD ADDRESSES
 787
 788 001064  172410                WCR:    172410
 789 001066  172412                BAR:    172412
 790 001070  172414                CSR:    172414
 791 001072  172416                BDR:    172416
 792 001074  000240                DRINL:  240
 793 001076  000124                VECADR: 124
 794 001100                              .BLKW   15.       ;TOTAL:16 LOCATIONS FOR VECTOR ADDRESSES
 795 001136  000124                DRINV:  124
 796 001140  000126                DRVS:   126
 797 001142  000000                EIR:    0               ;TEMP STORAGE FOR CSR #2 (ERROR IND. REG)
 798 001144  000000                BORW:   0
 799 001146  052525                NPR1:   52525
 800 001150  173000                DIOMEM: 173000
 801 001152  022406                INBUF:  XINBUF
 802 001154  023410                CHKBUF: XCHKBU
 803 001156  000000                BUFLEN: HALT
 804 001160  000000                LENCHK: HALT
 805 001162  000000                BRWAIT: HALT
 806 001164  000000                WCLEN:  HALT
 807 001166  000000                RDYCHK: HALT
 808 001170  177560                TKS:    177560
 809 001172  177562                TKB:    177562
 810 001174  177564                TPS:    177564
 811 001176  177566                TPB:    177566
 812 001200  000000                FLAG:   0
 813 001202  000000                FNCCNT: HALT
 814 001204  000000                INBUF1: HALT
 815 001206  000000                PASCNT: 0                       ;NUMBER OF PASSES COMPLETED: 16 BOARDS
 816 001210                              .BLKW   15.
 817 001246  000000                ERRCNT: 0                       ;ERROR COUNT TABLE:16 BOARDS
 818 001250                              .BLKW   15.
 819 001306  000000                TEMP:   0                       ;TEMPORARY STORAGE
 820 001310  000000                TEMP2:  0                       ;TEMPORARY STORAGE
 821 001312  000000                TIME:   0                       ;GENERAL PURPOSE TIMER
 822 001314  000000                CABLE:  0                       ;CABLE MODE TO BE DONE?:0-NO,1-YES
 823 001316  000000                HLTDET: 0                       ;FLAG THAT AN ERROR OCCURRED DURING PROGRAM PASS
 824 001320  000000                TESTNO: 0                       ;SPECIFIES TEST # FOR ERROR PRINT
 825 001322  000000                LOOP:   0                       ;GEN'L PURPOSE LOOP COUNTER
 826 001324  177570                DISPLAY:.WORD   DDISP
 827 001326  177570                SWR:    177570
 828
 829 001330  177560                $TKS:   177560
 830 001332  177562                $TKB:   177562
 831 001334  177564                $TPS:   177564
 832 001336  177566                $TPB:   177566
 833 001340     000                $NULL:  .BYTE   0
 834 001341     002                $FILLS: .BYTE   2
 835 001342     012                $FILLC: .BYTE   12
 836 001343     000                $TPFLG: .BYTE   0
 837 001344     207      377      377    $BELL:  .ASCIZ  <207><377><377>
```

```
     001347     000
838  001350     077                    $QUES:  .ASCII  /?/
839  001351     015                    $CRLF:  .ASCII  <15>
840  001352     012     000            $LF:    .ASCIZ  <12>
841  001354     377     377     000    $ENULL: .BYTE   -1,-1,0
842                                            .EVEN
843  001360  000000                     ANSWER: 0
844  001362  000000                     BDFAIL: 0
845
846  001364                             SUSWR:
848                                     .SBTTL  INITIALIZE THE COMMON TAGS
     001364  012706  022404                     MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
     001370  012737  017176  000034             MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
     001376  012737  000340  000036             MOV     #340,@#TRAPVEC+2;LEVEL 7
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
     001404  013746  000004                     MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
     001410  012737  001444  000004             MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
     001416  012737  177570  001326             MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
     001424  012737  177570  001324             MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
     001432  022777  177777  177666             CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
     001440  001012                             BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                        ;;AND  THE HARDWARE SWR IS NOT - -1
     001442  000403                             BR      65$             ;;BRANCH IF NO TIMEOUT
     001444  012716  001452            64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
     001450  000002                             RTI
     001452  012737  000176  001326     65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
     001460  012737  000174  001324             MOV     #DISPREG,DISPLAY
     001466  012637  000004            66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
849  001472  012737  001024  001010             MOV     #REGADR,PNTRAD  ;SETUP POINTERS IN BOARD TABLES
850  001500  012737  001076  001012             MOV     #VECADR,PNTVAD
851  001506  012737  001206  001014             MOV     #PASCNT,PNTPAS  ;
852  001514  012737  001246  001016             MOV     #ERRCNT,PNTERR
853
854  001522  012700  001206                     MOV     #PASCNT,R0      ;CLEAR PASS COUNT TABLE
855  001526  012701  000020                     MOV     #16.,R1
856  001532  005020                     1$:     CLR     (R0)+
857  001534  005301                             DEC     R1
858  001536  001375                             BNE     1$
859  001540  012700  001246                     MOV     #ERRCNT,R0      ;CLEAR ERROR COUNT TABLE
860  001544  012701  000020                     MOV     #16.,R1
861  001550  005020                     2$:     CLR     (R0)+
862  001552  005301                             DEC     R1
863  001554  001375                             BNE     2$
864  001556  005037  001020                     CLR     PRGCYC
865  001562  022737  000176  001326             CMP     #SWREG,SWR      ;IS SWREG USED
866  001570  001005                             BNE     BEGIN
867  001572  005737  000042                     TST     @#42            ;ARE WE IN AUTO MODE?
868  001576  001002                             BNE     BEGIN           ;IF SO, SKIP SWREG INPUT
869  001600  004737  014250                     JSR     PC,CNTLU        ;ALLOW SWREG TO BE LOADED
870  001604  000005                     BEGIN:  RESET
871  001606  005037  001316                     CLR     HLTDET          ;EVERY PASS CLEAR ERROR FLAG
872  001612  023737  001020  001022             CMP     PRGCYC,QTYBRD   ;HAS PROGRAM TESTED ALL BOARDS?
873  001620  001016                             BNE     BRDSET          ;NO,CONTINUE TO NEXT BOARD
874  001622  005037  001020                     CLR     PRGCYC          ;YES;INITIALIZE TESTING TO 1ST BOARD
875                                                                     ;OF MULTIPLE BOARD TABLE
```

```
                                        MOV     #REGADR,PNTRAD
                                        MOV     #VECADR,PNTVAD
                                        MOV     #PASCNT,PNTPAS
                                        MOV     #ERRCNT,PNTERR


                                        ;SET UP BOARD ADDRESSES OF CURRENT BOARD FROM
                                        ;MULTIPLE BOARD TABLE

                                BRDSET: INC     PRGCYC
                                        TST     @#42            ;ARE WE IN MANUAL MODE?
                                        BEQ     LOAD            ;YES;CHECK MULT BRD TABLE FOR NEXT BOARD ADDRESS
                                BEGAUT: MOV     DEFRAD,REGADR   ;NO;USE DEFAULT REGISTER ADDRESS
                                        MOV     DEFVAD,VECADR
                                        MOV     #1,QTYBRD
                                LOAD:   MOV     #WCR,R0 ;LOAD WRD CNT,BUS ADDRESS,STATUS,& DATA BUFFER ADDRESSES
                                        MOV     @PNTRAD,R1
                                2$:     MOV     R1,(R0)+        ;
                                        ADD     #2,R1           ;
                                        CMP     #BDR+2,R0       ;
                                        BNE     2$
                                        MOV     #DRINV,R0       ;LOAD INTERRUPT VECTOR ADDRESSES
                                        MOV     @PNTVAD,R1
                                3$:     MOV     R1,(R0)+        ;
                                        ADD     #2,R1
                                        CMP     #DRVS+2,R0
                                        BNE     3$
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF><CRLF>/*BRD ADDR.: /
                                64$:
                                        MOV     WCR,-(SP)       ;;SAVE WCR FOR TYPEOUT
                                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                        TYPF    ,67$            ;;TYPE ASCIZ STRING
                                        BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <CRLF>
                                66$:
                                1$:
                                        CLR     CABLE
                                        MOV     #BUFF,SP
                                        MOV     #340,@PSW       ;PROC. AT LEVEL #7
                                        MOV     T1STR,RETURN
                                        CLR     SCOPEF
                                        CLR     TESTNO
                                        CLR     ICOUNT
```

```
                        .SBTTL
                        .SBTTL          MAINTENANCE MODE TESTING
                        .SBTTL
```

```
                                        .SBTTL TEST 1 CAN ALL DR11W REG BE ADDRESSED WITHOUT ERROR?
                                ;;******************************************************************
                                ;       TEST 1 CAN ALL DR11-W REG BE ADDRESSED WITHOUT ERROR?
                                ;;******************************************************************


002066  000004                          SCOPE
002070  012737  002000  014122  T1STR:  MOV     #2000,ICOUNT    ;
002076  012737  002204  000004          MOV     #CPUERR,BUSERR  ;CPU ERROR  VECTOR TO CPUERR
002104  013746  001064          CKWCR:  MOV     WCR,-(SP)       ;SAVE ADDRESS FOR TYPE
002110  005777  176750                  TST     @WCR            ;ACCESS REGISTER
002114  005726                          TST     (SP)+           ;OK;READJUST STACK
002116  013746  001066          CKBAR:  MOV     BAR,-(SP)
002122  005777  176740                  TST     @BAR
002126  005726                          TST     (SP)+
002130  013746  001070          CKCSR:  MOV     CSR,-(SP)
002134  005777  176730                  TST     @CSR
002140  005726                          TST     (SP)+
002142  013746  001072          CKBDR:  MOV     BDR,-(SP)
002146  005777  176720                  TST     @BDR
002152  005726                          TST     (SP)+
002154  013746  001136          CKDRIN: MOV     DRINV,-(SP)
002160  013777  001140  176750        MOV     DRVS,@DRINV
002166  005726                          TST     (SP)+
002170  013746  001140          CKDRVS: MOV     DRVS,-(SP)
002174  005077  176740                  CLR     @DRVS
002200  005726                          TST     (SP)+
002202  000455                          BR      CKFIN
002204  022626                  CPUERR: CMP     (SP)+,(SP)+
002206  104401  002214                  TYPE    ,65$            ;;TYPE ASCIZ STRING
002212  000423                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/TESTNO 1-CPU ERROR WHILE ADRESSING /
002262                          64$:
002262  104402                          TYPOC
002264  104401  001354                  TYPE    ,$ENULL
002270  012737  000006  000004        MOV     #6,BUSERR       ;RESTORE #6 TO BUS ERROR VECTOR
002276  104401  002304                  TYPE    ,67$            ;;TYPE ASCIZ STRING
002302  000413                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <CRLF>/SKIPPING ALL TESTS /
002332                          66$:
002332  000137  013132                  JMP     END
002336  012737  000006  000004  CKFIN:  MOV     #6,BUSERR       ;RESTORE #6 TO BUS ERROR VECTOR
```

```
                                                .SBTTL TEST 2 ARE WE TESTING A DR11W OR A DR11B?
                                        ;;******************************************************************
                                        ;       TEST 2 ARE WE TESTING A DR11W OR A DR11B?
                                        ;;******************************************************************
002344  000004                                    SCOPE
002346  005037  014122                            CLR     ICOUNT
002352  005077  176512                            CLR     @CSR                    ;FORCE TO BE CSR #*
002356  052777  100000  176504                    BIS     #BIT15,@CSR
002364  017737  176500  001144                    MOV     @CSR,BORW
002372  032737  000001  001144                    BIT     #BIT0,BORW              ;ARE WE A B OR A W?
002400  001403                                    BEQ     1$                      ;WE ARE A B
002402  104401  015310                            TYPE    ,MSG9
002406  000402                                    BR      T3
002410  104401  015230                    1$:     TYPE    ,MSG8
```

```
                                        .SBTTL TEST 3 THAT DEVICE INIT CLEARS WCR
                                ;;*****************************************************************
                                ;       TEST 3  THAT DEVICE INIT CLEARS WCR
                                ;;*****************************************************************
002414  000004                T3:     SCOPE
002416  005077  176446                CLR     @CSR            ;FORCE TO BE CSR #1
002422  012737  000010  014122        MOV     #10,ICOUNT
002430  012777  177777  176426        MOV     #-1,@WCR        ;ALL ONES TO WCR
002436  004737  014144                JSR     PC,CKSWR
002442  052777  010000  176420        BIS     #BIT12,@CSR     ;DEVICE
002450  042777  010000  176412        BIC     #BIT12,@CSR     ;INIT
002456  005777  176402                TST     @WCR            ;IS Z BIT SET?
002462  001401                        BEQ     .+4             ;DID WCR GET CLEARED
002464  104000                        HLT                     ;WCR NOT CLEAR
```

```
                                                .SBTTL TEST 4 THAT DEVICE INIT CLEARS BAR
                                        ;;*****************************************************************
                                        ;       TEST 4 THAT DEVICE INIT CLEARS BAR
                                        ;;*****************************************************************
                                                SCOPE
002466  000004                                    CLR     @CSR            ;FORCE TO BE CSR #1
002470  005077  176374                           MOV     #-1,@BAR        ;ALL ONES TO BAR
002474  012777  177777  176364                   JSR     PC,CKSWR
002502  004737  014144                           BIS     #BIT12,@CSR     ;DEVICE
002506  052777  010000  176354                   BIC     #BIT12,@CSR     ;INIT
002514  042777  010000  176346                   TST     @BAR            ;IS Z BIT SET?
002522  005777  176340                           BEQ     .+4             ;DID BAR GET CLEARED
002526  001401                                   HLT                     ;BAR NO CLEAR
002530  104000
```

```
                                        .SBTTL TEST 5 THAT DEVICE INIT CLEARS BDR
                                ;;*****************************************************************
                                ;       TEST 5 THAT DEVICE INIT CLEARS BDR
                                ;;*****************************************************************
                                        SCOPE
002532  000004                          CLP     @CSR            ;FORCE TO BE CSR #1
002534  005077  176330                  MOV     #BIT12,@CSR     ;MAINT MODE
002540  012777  010000  176322          MOV     #10,ICOUNT
002546  012737  000010  014122          MOV     #-1,@BDR        ;ALL ONES TO BDR
002554  012777  177777  176310          JSR     PC,CKSWR
002562  004737  014144                  BIC     #BIT12,@CSR     ;DEVICE INIT
002566  042777  010000  176274          MOV     #BIT12,@CSR
002574  012777  010000  176266          TST     @BDR            ;CHECK FOR ZERO
002602  005777  176264                  BEQ     .+4
002606  001401                          HLT
002610  104000
```

```
                                        .SBTTL TEST 6 THAT DEVICE INIT CLEARS ALL R/W BITS IN THE CSR
                                ;;*****************************************************************
                                ;       TEST 6 THAT DEVICE INIT CLEARS ALL R/W BITS IN THE CSR
                                ;;*****************************************************************
002612  000004                        SCOPE
002614  012777  010000  176246        MOV     #BIT12,@CSR     ;MAINT MODE
002622  012737  000010  014122        MOV     #10,ICOUNT
002630  052777  010576  176232        BIS     #10576,@CSR     ;SET MAINT(12),CYCLE(08),IE(06)XBA17(05)
                                                                ;XBA16(04),FNCT03(03),FNCT02(02),FNCT1(01)
002636  004737  014144                JSR     PC,CKSWR
002642  042777  010000  176220        BIC     #BIT12,@CSR     ;DEVICE INIT
002650  012777  010000  176212        MOV     #BIT12,@CSR     ;MAINT MODE
002656  022777  010200  176204        CMP     #10200,@CSR     ;TEST FOR ZERO'S,AND READY BIT SET
002664  001401                        BEQ     .+4             ;WERE ALL R/W BITS CLEARED
002666  104000                        HLT
```

```
1037                                    .SBTTL TEST 7 DOES RESET CLEAR WCR?
1038                                ;;**********************************************************************
1039                                ;       TEST 7 DOES RESET CLEAR WCR?
1040                                ;;**********************************************************************
1041
1042 002670  000004                        SCOPE
1043 002672  005077  176172                CLR     @CSR            ;FORCE TO BE CSR #1
1044 002676  012737  000010  014122        MOV     #10,ICOUNT
1045 002704  012777  177777  176152        MOV     #-1,@WCR        ;ALL ONES TO WCR
1046 002712  004737  014144                JSR     PC,CKSWR
1047 002716  000005                        RESET                   ;INIT
1048 002720  005777  176140                TST     @WCR            ;LOOKING FOR Z-BIT TO SET
1049 002724  001401                        BEQ     .+4             ;DID WCR GET CLEARED?
1050 002726  104000                        HLT                     ;WCR NOT CLEAR
1051
1052
1053
```

```
                                        .SBTTL TEST 10 CAN ALL WCR BITS BE SET?
                                ;;**************************************************************************
                                ;       TEST 10 CAN ALL WCR BITS BE SET?
                                ;;**************************************************************************


002730  000004                        SCOPE
002732  005077  176132                CLR     @CSR            ;FORCE TO BE CSR #1
002736  012777  177777  176120        MOV     #-1,@WCR        ;SET ALL BITS IN WCR
002744  022777  177777  176112        CMP     #-1,@WCR        ;LOOKING FOR 2-BIT TO SET
002752  001401                        BEQ     .+4             ;SEE IF ALL BITS GOT SET
002754  104000                        HLT                     ;ALL BITS AREN'T SET
```

```
                                        .SBTTL TEST 11 DOES RESET CLEAR BAR?
                                ;;**************************************************************************
                                ;       TEST 11 DOES RESET CLEAR BAR?
                                ;;**************************************************************************


002756  000004                          SCOPE
002760  005077  176104                  CLR     @CSR            ;FORCE TO BE CSR #1
002764  012777  177777  176074          MOV     #-1,@BAR        ;ALL ONES TO BAR
002772  004737  014144                  JSR     PC,CKSWR
002776  000005                          RESET                   ;INIT
003000  005777  176062                  TST     @BAR            ;LOOKING FOR Z-BIT TO SET
003004  001401                          BEQ     .+4             ;DID BAR GET CLEARED?
003006  104000                          HLT                     ;BAR NOT CLEAR
```

```
                                                .SBTTL TEST 12 CAN BITS 15-01 IN BAR BE SET?
                                        ;;**************************************************************************
                                        ;       TEST 12 CAN BITS 15-01 IN BAR BE SET?
                                        ;;**************************************************************************

003010  000004                                  SCOPE
003012  012777  177776  176046                  MOV     #-2,@BAR        ;SET BITS 15-01 IN BAR
003020  022777  177776  176040                  CMP     #-2,@BAR        ;LOOKING FOR Z-BIT TO SET
003026  001401                                  BEQ     .+4             ;SEE IF BITS 15-01 GOT SET

003030  104000                                  HLT
```

```
1101                                    .SBTTL TEST 13 TEST THAT FNCT1 CAN BE SET AND CLEARED
1102                                ;;**********************************************************************
1103                                ;       TEST 13 TEST THAT FNCT1 CAN BE SET AND CLEARED
1104                                ;;**********************************************************************
1105
1106
1107 003032  000004                         SCOPE
1108 003034  005077  176030                 CLR     @CSR            ;FORCE TO BE CSR #1
1109 003040  012777  010000  176022         MOV     #BIT12,@CSR     ;SET MAINT MODE
1110 003046  012737  002000  014122         MOV     #2000,ICOUNT
1111 003054  052777  000002  176006         BIS     #BIT1,@CSR      ;SET FNCT1
1112 003062  032777  000002  176000         BIT     #BIT1,@CSR      ;TEST FNCT1
1113 003070  001001                         BNE     .+4             ;IS IT SET?
1114 003072  104000                         HLT                     ;FNCT1 IS CLEAR
1115 003074  042777  000002  175766         BIC     #BIT1,@CSR      ;CLEAR FNCT1
1116 003102  032777  000002  175760         BIT     #BIT1,@CSR      ;TEST FNCT1
1117 003110  001401                         BEQ     .+4             ;WAS IT CLEAR
1118 003112  104000                         HLT                     ;FNCT1 WAS SET
1119
1120
1121
```

```
                                        .SBTTL TEST 14 TEST THAT FNCT2 CAN BE SET AND CLEARED
                                ;;******************************************************************
                                ;       TEST 14 TEST THAT FNCT2 CAN BE SET AND CLEARED
                                ;;******************************************************************


003114  000004                          SCOPE
003116  005077  175746                  CLR     @CSR            ;FORCE TO BE CSR #1
003122  012777  010000  175740          MOV     #BIT12,@CSR     ;SET MAINT MODE
003130  052777  000004  175732          BIS     #BIT2,@CSR      ;SET FNCT2
003136  032777  000004  175724          BIT     #BIT2,@CSR      ;TEST FNCT2
003144  001001                          BNE     .+4             ;IS IT SET?
003146  104000                          HLT                     ;FNCT2 IS CLEAR
003150  042777  000004  175712          BIC     #BIT2,@CSR      ;CLEAR FNCT2
003156  032777  000004  175704          BIT     #BIT2,@CSR      ;TEST FNCT2
003164  001401                          BEQ     .+4             ;WAS IT CLEAR?
003166  10400C                          HLT                     ;FNCT2 WAS SET
```

```
                                        .SBTTL TEST 15 TEST THAT FNCT3 CAN BE SET AND CLEARED
                                ;;*****************************************************************
                                ;       TEST 15 TEST THAT FNCT3 CAN BE SET AND CLEARED
                                ;;*****************************************************************

003170  000004                          SCOPE
003172  005077  175672                  CLR     @CSR            ;FORCE TO BE CSR #1
003176  012777  010000  175664          MOV     #BIT12,@CSR     ;SET MAINT MODE
003204  052777  000010  175656          BIS     #BIT3,@CSR      ;SET FNCT3
003212  032777  000010  175650          BIT     #BIT3,@CSR      ;TEST FNCT3
003220  001001                          BNE     .+4             ;IS IT SET?
003222  104000                          HLT                     ;FNCT3 IS CLEAR
003224  042777  000010  175636          BIC     #BIT3,@CSR      ;CLEAR FNCT3
003232  032777  000010  175630          BIT     #BIT3,@CSR      ;TEST FNCT3
003240  001401                          BEQ     .+4             ;WAS IT CLEAR?
003242  104000                          HLT                     ;FNCT3 WAS SET
```

```
                                        .SBTTL TEST 16 TEST THAT XBA16 CAN BE SET AND CLEARED
                                ;;**************************************************************
                                ;       TEST 16 TEST THAT XBA16 CAN BE SET AND CLEARED
                                ;;**************************************************************
003244  000004                          SCOPE
003246  005077  175616                  CLR     @CSR            ;FORCE TO BE CSR #1
003252  012777  010000  175610          MOV     #BIT12,@CSR     ;SET MAINT MODE
003260  052777  000020  175602          BIS     #BIT4,@CSR      ;SET XBA16
003266  032777  000020  175574          BIT     #BIT4,@CSR      ;TEST XBA16
003274  001001                          BNE     .+4             ;IS IT SET?
003276  104000                          HLT                     ;XBA16 IS CLEAR
003300  042777  000020  175562          BIC     #BIT4,@CSR      ;CLEAR XBA16
003306  032777  000020  175554          BIT     #BIT4,@CSR      ;TEST XBA16
003314  001401                          BEQ     .+4             ;IS IT CLEAR
003316  104000                          HLT                     ;XBA16 WAS SET
```

```
1180                                    .SBTTL TEST 17 TEST THAT XBA17 CAN BE SET AND CLEARED
1181                                ;;**********************************************************************
1182                                ;       TEST 17 TEST THAT XBA17 CAN BE SET AND CLEARED
1183                                ;;**********************************************************************
1184 003320  000004                        SCOPE
1185 003322  005077  175542                CLR     @CSR            ;FORCE TO BE CSR #1
1186 003326  012777  010000  175534        MOV     #BIT12,@CSR     ;SET MAINT MODE
1187 003334  052777  000040  175526        BIS     #BIT5,@CSR      ;SET XBA17
1188 003342  032777  000040  175520        BIT     #BIT5,@CSR      ;TEST XBA17
1189 003350  001001                        BNE     .+4             ;IS IT SET?
1190 003352  104000                        HLT                     ;XBA17 IS CLEAR
1191 003354  042777  000040  175506        BIC     #BIT5,@CSR      ;CLEAR XBA17
1192 003362  032777  000040  175500        BIT     #BIT5,@CSR      ;TEST XBA17
1193 003370  001401                        BEQ     .+4             ;IS IT CLEAR?
1194 003372  104000                        HLT                     ;XBA17 WAS SET
1195
```

```
1197                                            .SBTTL TEST 20 TEST THAT IE CAN BE SET AND CLEARED
1198                                    ;;**************************************************************
1199                                    ;       TEST 20 TEST THAT IE CAN BE SET AND CLEARED
1200                                    ;;**************************************************************
1201 003374  000004                            SCOPE
1202 003376  005077  175466                    CLR     @CSR            ;FORCE TO BE CSR #1
1203 003402  012777  010000  175460            MOV     #BIT12,@CSR     ;SET MAINT MODE
1204 003410  052777  000100  175452            BIS     #BIT6,@CSR      ;SET IE
1205 003416  032777  000100  175444            BIT     #BIT6,@CSR      ;TEST IE
1206 003424  001001                            BNE     .+4             ;IS IT SET?
1207 003426  104000                            HLT                     ;IE IS CLEAR
1208 003430  042777  000100  175432            BIC     #BIT6,@CSR      ;CLEAR IE
1209 003436  032777  000100  175424            BIT     #BIT6,@CSR      ;TEST IE
1210 003444  001401                            BEQ     .+4             ;IS IT CLEAR?
1211 003446  104000                            HLT                     ;IE WAS SET
1212
```

```
                                        .SBTTL TEST 21 TEST THAT CYCLE CAN BE SET AND CLEARED
                                ;;**************************************************************
                                ;       TEST 21 TEST THAT CYCLE CAN BE SET AND CLEARED
                                ;;**************************************************************
003450  000004                          SCOPE
003452  005077  175412                  CLR     @CSR            ;FORCE TO BE CSR #1
003456  012777  010000  175404          MOV     #BIT12,@CSR     ;SET MAINT MODE
003464  052777  000400  175376          BIS     #BIT8,@CSR      ;SET CYCLE
003472  032777  000400  175370          BIT     #BIT8,@CSR      ;TEST CYCLE
003500  001001                          BNE     .+4             ;IS IT SET?
003502  104000                          HLT                     ;CYCLE WAS CLEAR
003504  042777  000400  175356          BIC     #BIT8,@CSR      ;CLEAR CYCLE
003512  032777  000400  175350          BIT     #BIT8,@CSR      ;TEST CYCLE
003520  001401                          BEQ     .+4             ;IS IT CLEAR?
003522  104000                          HLT                     ;CYCLE WAS SET
```

```
                                        .SBTTL TEST 22 TEST THAT MAINT CAN BE SET AND CLEARED
                                ;;**********************************************************************
                                ;       TEST 22 TEST THAT MAINT CAN BE SET AND CLEARED
                                ;;**********************************************************************
003524  000004                          SCOPE
003526  005077  175336                  CLR     @CSR            ;FORCE TO BE CSR #1
003532  012777  010000  175330          MOV     #BIT12,@CSR     ;SET MAINT
003540  032777  010000  175322          BIT     #BIT12,@CSR     ;TEST MAINT
003546  001001                          BNE     .+4             ;IS IT SET?
003550  104000                          HLT                     ;MAINT WAS CLEAR
003552  042777  010000  175310          BIC     #BIT12,@CSR     ;CLEAR MAINT
003560  032777  010000  175302          BIT     #BIT12,@CSR     ;TEST MAINT
003566  001401                          BEQ     .+4             ;IS MAINT CLEAR?
003570  104000                          HLT                     ;MAINT WAS SET

003572  001401                          BEQ     .+4             ;IS IT CLEAR?
003574  104000                          HLT                     ;CYCLE WAS SET
```

```
                                        .SBTTL TEST 23 THAT ALL CSR R/W BITS CAN BE SET AND CLEARED
                                ;;******************************************************************************
                                ;       TEST 23 TEST THAT ALL CSR R/W BITS CAN BE SET AND CLEARED
                                ;       MAINT MODE(INTERNAL WRAP-AROUND)
                                ;;******************************************************************************
                                        SCOPE
003576  000004                  
003600  005077  175264          
003604  012777  010000  175256  
003612  052777  020576  175250  
                                ;                                       XBA16(04),FNCT3(03),FUNCT2(02),FNCT1(01)
003620  032777  000002  175242  
003626  001001                  
003630  104000                  
003632  032777  000004  175230  
003640  001001                  
003642  104000                  
003644  032777  000010  175216  
003652  001001                  
003654  104000                  
003656  032777  000020  175204  
003664  001001                  
003666  104000                  
003670  032777  000040  175172  
003676  001001                  
003700  104000                  
003702  032777  000100  175160  
003710  001001                  
003712  104000                  
003714  032777  000400  175146  
003722  001001                  
003724  104000                  
003726  032777  010000  175134  
003734  001001                  
003736  104000                  
```

1250							
1251							
1252							
1253							
1254							
1255	003576	000004				SCOPE	
1256	003600	005077	175264		CLR	@CSR	;FORCE TO BE CSR #1
1257	003604	012777	010000	175256	MOV	#BIT12,@CSR	;MAINT MODE
1258	003612	052777	020576	175250	BIS	#20576,@CSR	;SET FOLLOWING:MAINT(12), CYCLE(08),IE(06),XBA17(05),
1259							; XBA16(04),FNCT3(03),FUNCT2(02),FNCT1(01)
1260	003620	032777	000002	175242	BIT	#BIT1,@CSR	;TEST FNCT1
1261	003626	001001			BNE	.+4	;IS IT SET?
1262	003630	104000			HLT		;FNCT1 IS CLEAR
1263	003632	032777	000004	175230	BIT	#BIT2,@CSR	;TEST FNCT2
1264	003640	001001			BNE	.+4	;IS IT SET?
1265	003642	104000			HLT		;FNCT2 IS CLEAR
1266	003644	032777	000010	175216	BIT	#BIT3,@CSR	;TEST FNCT3
1267	003652	001001			BNE	.+4	;IS IT SET?
1268	003654	104000			HLT		;FNCT3 IS CLEAR
1269	003656	032777	000020	175204	BIT	#BIT4,@CSR	;TEST XBA16
1270	003664	001001			BNE	.+4	;IS IT SET?
1271	003666	104000			HLT		;XBA16 IS CLEAR
1272	003670	032777	000040	175172	BIT	#BIT5,@CSR	;TEST XBA17
1273	003676	001001			BNE	.+4	;IS IT SET?
1274	003700	104000			HLT		;XBA17 IS CLEAR
1275	003702	032777	000100	175160	BIT	#BIT6,@CSR	;TEST IE
1276	003710	001001			BNE	.+4	;IS IT SET?
1277	003712	104000			HLT		;IE IS CLEAR
1278	003714	032777	000400	175146	BIT	#BIT8,@CSR	;TEST CYCLE
1279	003722	001001			BNE	.+4	;IS CYCLE SET?
1280	003724	104000			HLT		;CYCLE IS CLEAR
1281	003726	032777	010000	175134	BIT	#BIT12,@CSR	;TEST MAINT
1282	003734	001001			BNE	.+4	;IS MAINT SET?
1283	003736	104000			HLT		;MAINT IS CLEAR

```
1285 003740  042777  030576  175122          BIC     #30576,@CSR     ;CLEAR ALL R/W BITS IN CSR
1286 003746  032777  000002  175114          BIT     #BIT1,@CSR      ;TEST FNCT1
1287 003754  001401                          BEQ     .+4             ;IS FNCT1 CLEAR?
1288 003756  104000                          HLT                     ;FNCT1 IS SET
1289 003760  032777  000004  175102          BIT     #BIT2,@CSR      ;TEST FNCT2
1290 003766  001401                          BEQ     .+4             ;IS FNCT2 CLEAR?
1291 003770  104000                          HLT                     ;FNCT2 IS SET
1292 003772  032777  000010  175070          BIT     #BIT3,@CSR      ;TEST FNCT3
1293 004000  001401                          BEQ     .+4             ;IS FNCT3 CLEAR?
1294 004002  104000                          HLT                     ;FNCT3 IS SET
1295 004004  032777  000020  175056          BIT     #BIT4,@CSR      ;TEST XBA16
1296 004012  001401                          BEQ     .+4             ;IS XBA16 CLEAR
1297 004014  104000                          HLT                     ;XBA16 IS SET
1298 004016  032777  000040  175044          BIT     #BIT5,@CSR      ;TEST XBA17
1299 004024  001401                          BEQ     .+4             ;IS XBA17 CLEAR?
1300 004026  104000                          HLT                     ;XBA17 IS SET
1301 004030  032777  000100  175032          BIT     #BIT6,@CSR      ;TEST IE
1302 004036  001401                          BEQ     .+4             ;IS IE CLEAR?
1303 004040  104000                          HLT                     ;IE IS SET
1304 004042  032777  000400  175020          BIT     #BIT8,@CSR      ;TEST CYCLE
1305 004050  001401                          BEQ     .+4             ;IS CYCLE CLEAR?
1306 004052  104000                          HLT                     ;CYCLE IS SET
1307 004054  032777  010000  175006          BIT     #BIT12,@CSR     ;TEST MAINT
1308 004062  001401                          BEQ     .+4             ;IS MAINT CLEAR?
1309 004064  104000                          HLT                     ;MAINT IS SET
1310
```

```
1312                                            .SBTTL TEST 24 ALL R/W BITS IN CSR CAN BE SET AND RESET TO ZERO
1313                                    ;;*****************************************************************************
1314                                    ;       TEST 24 ALL R/W BITS IN CSR CAN BE SET AND RESET TO ZERO, THAT READY
1315                                    ;       IS SET, NEX IS CLEAR,  GO IS READ AS A 0,AND ERROR  IS CLEAR.
1316                                    ;       MAINT MODE(INTERNAL WRAP-AROUND)
1317                                    ;;*****************************************************************************
1318 004066  000004                            SCOPE
1319 004070  005077  174774                    CLR     @CSR                    ;FORCE TO BE CSR #1
1320 004074  012777  010000  174766            MOV     #BIT12,@CSR             ;MAINT MODE
1321 004102  012737  000010  014122            MOV     #10,ICOUNT
1322 004110  052777  010576  174752            BIS     #10576,@CSR             ;SET FOLLOWING:  MAINT(12),CYCLE(08),IE(06),XBA17(05),
1323                                                                            ;                XBA16(04),FNCT3(03),FNCT2(02),FNCT1(01)
1324 004116  017701  174746                    MOV     @CSR,R1                 ;MOVE (CSR) TO R1
1325 004122  052701  167201                    BIS     #167201,R1              ;SETS BITS IN R1 THAT WERE NOT SET IN CSR
1326 004126  005201                            INC     R1                      ;R1 SHOULD GO FROM -1 TO ZERO
1327 004130  001401                            BEQ     .+4                     ;WERE ALL CSR R/W BITS SET?
1328 004132  104000                            HLT                             ;NOT ALL BITS WERE SET
1329 004134  004737  014144                    JSR     PC,CKSWR
1330 004140  000005                            RESET                           ;CLEAR ALL CSR R/W BITS
1331 004142  017701  174722                    MOV     @CSR,R1                 ;MOVE (CSR) TO R1
1332 004146  042701  127000                    BIC     #127000,R1              ;CLEAR ERROR,ATTN,DSTATA,DSTATB,DSTATC
1333 004152  022701  000200                    CMP     #200,R1                 ;TEST FOR READY
1334 004156  001401                            BEQ     .+4
1335 004160  104000                            HLT                             ;RESET DIDN'T LEAVE CSR AS IT SHOULD HAVE
1336
```

```
1338                                    .SBTTL TEST 25 CSR AND EIR BIT0 TEST
1339                                ;;**********************************************************************
1340                                ;       TEST 25 CSR AND EIR BIT0 TEST
1341                                ;       TEST THAT BIT0 OF CSR READS AS A ZERO
1342                                ;       TEST THAT BIT0 OF EIR READS AS A ONE
1343                                ;;**********************************************************************
1344 004162  000004                        SCOPE
1345 004164  032737  000001  001144        BIT     #BIT0,BORW
1346 004172  001426                        BEQ     T26
1347 004174  005077  174670                CLR     @CSR                ;FORCE TO BE CSR
1348 004200  032777  000001  174662        BIT     #BIT0,@CSR          ;IS BIT0 A ZERO?
1349 004206  001401                        BEQ     .+4                 ;YES IT IS A ZERO
1350 004210  104000                        HLT                         ;BIT0 READ AS A ONE
1351 004212  012777  100000  174650        MOV     #BIT15,@CSR         ;SET BIT15 TO GO TO EIR
1352 004220  032777  000001  174642        BIT     #BIT0,@CSR          ;IS BIT0 A ONE?
1353 004226  001001                        BNE     .+4                 ;YES IT IS A ONE
1354 004230  104000                        HLT                         ;BIT0 READS AS A ZERO
1355 004232  005077  174632                CLR     @CSR                ;RETURN TO CSR
1356 004236  032777  000001  174624        BIT     #BIT0,@CSR          ;DID WE GET BACK
1357 004244  001401                        BEQ     .+4                 ;YES
1358 004246  104000                        HLT                         ;NO WE ARE STUCK IN EIR
1359
```

```
                        .SBTTL TEST 26 THAT ATTN CAN BE SET VIA F2 & ERROR SETS
                ;;*****************************************************************
                ;       TEST 26 TEST THAT ATTN CAN BE SET VIA F2 & ERROR SETS;
                ;       TEST THAT ATTN,AND THEREFORE ERROR WILL CLEAR
                ;       MAINTENANCE MODE (INTERNAL WRAP-AROUND)
                ;;*****************************************************************
004250  000004         T26:   SCOPE
004252  032737  000001  001144        BIT     #BIT0,BORW
004260  001444                        BEQ     T27             ;IF A 'B' SKIP TEST
004262  012737  002000  014122        MOV     #2000,ICOUNT
004270  005077  174574                CLR     @CSR            ;FORCE TO BE CSR #1
004274  012777  010000  174566        MOV     #BIT12,@CSR     ;MAINT
004302  022777  010200  174560        CMP     #10200,@CSR     ;ONLY READY AND MAINT
004310  001401                        BEQ     .+4
004312  104000                        HLT                     ;SOMETHING OTHER THAN READY&MAINT
004314  112777  000004  174546        MOVB    #BIT2,@CSR      ;SET F2
004322  032777  020000  174540        BIT     #BIT13,@CSR     ;TEST ATTN
004330  001001                        BNE     .+4             ;IS IT SET?
004332  104000                        HLT                     ;ATTN WAS CLEAR
004334  042777  020004  174526        BIC     #20004,@CSR     ;CLEAR ATTN & F2
004342  032777  020000  174520        BIT     #BIT13,@CSR     ;TEST ATTN
004350  001401                        BEQ     .+4             ;IS ATTN CLEAR?
004352  104000                        HLT                     ;ATTN WAS SET
004354  017705  174510                MOV     @CSR,R5         ;SAVE CSR #1
004360  005705                        TST     R5              ;IS ERROR CLEAR?
004362  100001                        BPL     .+4
004364  104000                        HLT                     ;ERROR IS STILL SET
004366  005077  174476                CLR     @CSR            ;RETURN TO CSR #1
```

```
                                        .SBTTL TEST 27 CAN WCR HOLD ALTERNATE ONES AND ZEROS
                                ;;******************************************************************
                                ;       TEST 27 CAN WCR HOLD ALTERNATE ONE'S AND ZERO'S
                                ;;******************************************************************
004372  000004                T27:    SCOPE
004374  005077  174470                CLR     @CSR            ;FORCE TO BE CSR #1
004400  012737  002000  014122        MOV     #2000,ICOUNT
004406  012777  052525  174450        MOV     #052525,@WCR    ;ALT 0'S AND 1'S TO WCR
004414  022777  052525  174442        CMP     #052525,@WCR    ;LOOKING FOR Z-BIT TO SET
004422  001401                        BEQ     .+4             ;DOES WCR HAVE THE CORRECT PATTERN?
004424  104000                        HLT                     ;WCR DOESN'T HAVE THE CORRECT PATTERN
004426  012777  125252  174430        MOV     #125252,@WCR    ;ALT 1'S AND 0'S TO WCR
004434  022777  125252  174422        CMP     #125252,@WCR    ;LOOKING FOR Z-BIT TO SET
004442  001401                        BEQ     .+4             ;DOES WCR HAVE THE CORRECT PATTERN?
004444  104000                        HLT                     ;WCR DOESN'T HAVE THE CORRECT PATTERN
```

```
                                        .SBTTL TEST 30 TEST THAT BA00 READS AS A ZERO
                                ;;******************************************************************
                                ;       TEST 30 TEST THAT BA00 READS AS A ZERO
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;;******************************************************************
004446  000004                          SCOPE
004450  005077  174414                  CLR     @CSR            ;FORCE TO BE CSR #1
004454  012737  002000  014122          MOV     #2000,ICOUNT
004462  012777  010000  174400          MOV     #BIT12,@CSR     ;MAINT MODE
004470  000240                          NOP
004472  000240                          NOP
004474  105777  174370                  TSTB    @CSR            ;IN MAINTENANCE MODE TESTING:READY
                                                                ;CONTROLS LEVEL OF BA00.MUST BE SET
                                                                ;FOR BA00=0.ALSO,READY SET ALLOWS CSR
                                                                ;TO BE READ.READING CSR WILL ENABLE
                                                                ;CLOCKING OF 0 TO BA00 THRU REGISTER
                                                                ;IN BOARD HARDWARE
004500  100401                          BMI     .+4
004502  104000                          HLT                     ;READY NOT SET
004504  032777  000001  174354          BIT     #BIT0,@BAR      ;TEST BIT 0 OF BAR
004512  001401                          BEQ     .+4             ;IS IT CLEAR?
004514  104000                          HLT                     ;BA00 IS SET
```

```
                                        .SBTTL TEST 31 CAN BAR HOLD ALTERNATE ONES AND ZEROS
                                ;;*****************************************************************
                                ;       TEST 31 CAN BAR HOLD ALTERNATE ONE'S AND ZERO'S
                                ;       MAINT MODE(INTERN. WRAP-AROUND)
                                ;;*****************************************************************
004516  000004                        SCOPE
004520  005077  174344                CLR     @CSR            ;FORCE TO BE CSR #1
004524  012777  010000  174336        MOV     #BIT12,@CSR     ;MAINT MODE
004532  012777  052524  174326        MOV     #052524,@BAR    ;ALT 0'S AND 1'S TO BAR
004540  005777  174324                TST     @CSR
004544  022777  052524  174314        CMP     #052524,@BAR    ;LOOKING FOR Z-BIT TO SET
004552  001401                        BEQ     .+4             ;DOES BAR HAVE THE CORRECT PATTERN?
004554  104000                        HLT                     ;BAR DOESN'T HAVE THE CORRECT PATTERN
004556  012777  125252  174302        MOV     #125252,@BAR    ;ALT 1'S AND 0'S TO BAR
004564  005777  174300                TST     @CSR
004570  022777  125252  174270        CMP     #125252,@BAR    ;LOOKING FOR Z BIT TO SET
004576  001401                        BEQ     .+4
004600  104000                        HLT                     ;INCORRECT PATTERN
```

```
1451                                    .SBTTL TEST 32 INCREMENTING PATTERN TO WRAP-AROUND IN WCR
1452                                ;;**********************************************************************
1453                                ;       TEST 32 INCREMENTING PATTERN TO WRAP-AROUND IN WCR
1454                                ;;**********************************************************************
1455 004602  000004                        SCOPE
1456 004604  005077  174260                CLR     @CSR            ;FORCE TO BE CSR #1
1457 004610  005037  014122                CLR     ICOUNT
1458 004614  005001                        CLR     R1              ;SET-UP
1459 004616  005077  174242                CLR     @WCR            ;SET-UP
1460 004622  020177  174236        INCWC:  CMP     R1,@WCR         ;SEE IF THEY ARE EQUAL
1461 004626  001401                        BEQ     .+4             ;ARE THEY EQUAL?
1462 004630  104000                        HLT                     ;THEY'RE NOT EQUAL
1463 004632  005277  174226                INC     @WCR            ;GET NEXT NUMBER
1464 004636  005201                        INC     R1              ;GET NEXT NUMBER
1465 004640  001370                        BNE     INCWC           ;DONE WITH TEST?  IF NOT CONTINUE
1466
```

```
                                        .SBTTL TEST 33 INCREMENTING PATTERN TO WRAP-AROUND IN BAR
                                ;;*****************************************************************
                                ;       TEST 33 INCREMENTING PATTERN TO WRAP-AROUND IN BAR
                                ;       MAINT MODE(INTERN. WRAP-AROUND)
                                ;;*****************************************************************
004642  000004                          SCOPE
004644  005001                          CLR     R1              ;SET-UP
004646  005077  174214                  CLR     @BAR            ;SET-UP
004652  020177  174210          INCBA:  CMP     R1,@BAR         ;SEE IF THEY ARE EQUAL
004656  001401                          BEQ     .+4             ;ARE THEY EQUAL?
004660  104000                          HLT                     ;THEY'RE NOT EQUAL
004662  062777  000002  174176          ADD     #2,@BAR         ;GET NEXT NUMBER
004670  062701  000002                  ADD     #2,R1           ;GET NEXT NUMBER
004674  001366                          BNE     INCBA           ;DONE WITH TEST?  IF NOT CONTINUE
```

```
                                        .SBTTL TEST 34 TEST THAT FNCT BITS CONTROL DSTAT BITS
                                ;;*****************************************************************
                                ;       TEST 34 TEST THAT FNCT BITS CONTROL DSTAT BITS
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;;*****************************************************************
004676  000004                  X5:     SCOPE
004700  005077  174164                  CLR     @CSR            ;CLR FUNCT BITS AND FORCE TO BE CSR #1
004704  012777  010000  174156          MOV     #10000,@CSR     ;MAINT MODE
004712  032777  000010  174150          BIT     #16,@CSR        ;CHECK FUNCTION BITS
004720  001401                          BEQ     .+4             ;FUNCTION BITS CLEAR?
004722  104000                          HLT                     ;FUNCTION BITS NOT CLEAR
004724  052777  000002  174136          BIS     #BIT1,@CSR      ;SET FNCT1
004732  032777  001000  174130          BIT     #BIT9,@CSR      ;CHECK DSTAT C
004740  001001                          BNE     .+4             ;IS IT SET?
004742  104000                          HLT                     ;DSTAT C IS CLEAR
004744  032777  006000  174116          BIT     #6000,@CSR      ;CHECK THAT DSTAT A AND DSTAT B ARE CLEAR
004752  001401                          BEQ     .+4             ;ARE THEY CLEAR?
004754  104000                          HLT                     ;DSTAT A AND/OR DSTAT B IS SET
004756  012777  010000  174104          MOV     #10000,@CSR     ;MAINT MODE
004764  052777  000004  174076          BIS     #BIT2,@CSR      ;SET FNCT2
004772  032777  002000  174070          BIT     #BIT10,@CSR     ;CHECK DSTAT B
005000  001001                          BNE     .+4             ;IS IT SET?
005002  104000                          HLT                     ;DSTAT B IS CLEAR
005004  032777  005000  174056          BIT     #5000,@CSR      ;CHECK THAT DSTAT A AND DSTAT C ARE CLEAR
005012  001401                          BEQ     .+4             ;ARE THEY CLEAR?
005014  104000                          HLT                     ;DSTAT A AND/OR DSTAT B IS SET
005016  012777  010000  174044          MOV     #10000,@CSR     ;MAINT MODE
005024  052777  000010  174036          BIS     #BIT3,@CSR      ;SET FNCT3
005032  032777  004000  174030          BIT     #BIT11,@CSR     ;CHECK DSTAT A
005040  001001                          BNE     .+4             ;IS IT SET?
005042  104000                          HLT                     ;DSTAT A IS CLEAR
005044  032777  003000  174016          BIT     #3000,@CSR      ;CHECK THAT DSTAT B AND DSTAT C ARE CLEAR
005052  001401                          BEQ     .+4             ;ARE THEY CLEAR?
005054  104000                          HLT                     ;DSTAT B AND/OR DSTAT C IS SET
```

```
                                        .SBTTL TEST 35 TEST THAT RESET CLEARS BDR
                                ;;*****************************************************************
                                ;       TEST 35 TEST THAT RESET CLEARS BDR
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;;*****************************************************************
005056  000004                          SCOPE
005060  005077  174004                  CLR     @CSR            ;FORCE TO BE CSR #1
005064  012737  000010  014122          MOV     #10,ICOUNT
005072  012777  177777  173772          MOV     #-1,@BDR        ;ALL ONES TO BDR
005100  004737  014144                  JSR     PC,CKSWR
005104  000005                          RESET                   ;INIT
005106  052777  010000  173754          BIS     #10000,@CSR     ;MAINT MODE
005114  005777  173752                  TST     @BDR            ;LOOKING FOR Z-BIT TO SET
005120  001401                          BEQ     .+4             ;DID BDR GET CLEARED?
005122  104000                          HLT                     ;BDR NOT CLEAR
```

```
                                        .SBTTL TEST 36 TEST THAT ALL BDR BITS CAN BE SET
                                ;;*****************************************************************
                                ;       TEST 36 TEST THAT ALL BDR BITS CAN BE SET
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND
                                ;;*****************************************************************
005124  000004                          SCOPE
005126  005077  173736                  CLR     @CSR            ;FORCE TO BE CSR #1
005132  012737  002000  014122          MOV     #2000,ICOUNT
005140  012777  010000  173722          MOV     #10000,@CSR     ;MAINT MODE
005146  012777  177777  173716          MOV     #-1,@BDR        ;SET ALL BITS IN BDR
005154  022777  177777  173710          CMP     #-1,@BDR        ;LOOKING FOR Z-BIT TO SET
005162  001401                          BEQ     .+4             ;SEE IF ALL BITS GOT SET
005164  104000                          HLT                     ;ALL BDR BITS AREN'T SET
```

```
                                        .SBTTL TEST 37 TEST THAT BDR CAN HOLD ALTERNATE ONES AND ZEROS
                                ;;******************************************************************************
                                ;       TEST 37 TEST THAT BDR CAN HOLD ALTERNATE ONE'S AND ZERO'S
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;;******************************************************************************
005166  000004                        SCOPE
005170  005077  173674                CLR     @CSR                    ;FORCE TO BE CSR #1
005174  012777  010000  173666        MOV     #10000,@CSR             ;MAINT MODE
005202  012777  052525  173662        MOV     #052525,@BDR            ;ALT 0'S AND 1'S TO BDR
005210  022777  052525  173654        CMP     #052525,@BDR            ;LOOKING FOR Z-BIT TO SET
005216  001401                        BEQ     .+4                     ;DOES BDR HAVE THE CORRECT PATTERN?
005220  104000                        HLT                             ;BDR IS WRONG
005222  012777  125252  173642        MOV     #125252,@BDR            ;ALT 1'S AND 0'S TO BDR
005230  022777  125252  173634        CMP     #125252,@BDR            ;LOOKING FOR Z-BIT TO SET
005236  001401                        BEQ     .+4                     ;DOES BDR HAVE THE CORRECT PATTERN
005240  104000                        HLT                             ;BDR IS WRONG
```

```
                                        .SBTTL TEST 40 THAT GOING TO EIR BLOCKS DATA XFERS FROM ODR TO IDR
                                ;;**********************************************************************
                                ;       TEST 40 TEST THAT GOING TO EIR BLOCKS DATA TRANSFER FROM
                                ;       ODR TO IDR
                                ;;**********************************************************************
005242  000004                        SCOPE
005244  032737  000001  001144        BIT     #BIT0,@BORW
005252  001442                        BEQ     T41
005254  005077  173610                CLR     @CSR            ;FORCE TO BE CSR
005260  012737  002000  014122        MOV     #2000,ICOUNT    ;
005266  012777  010000  173574        MOV     #BIT12,@CSR     ;SET MAINT MODE
005274  012777  052525  173570        MOV     #52525,@BDR     ;SET ALT 0'S AND 1'S TO BDR
005302  022777  052525  173562        CMP     #52525,@BDR     ;LOOK FOR CORRECT PATTERN
005310  001401                        BEQ     .+4             ;BDR IS GOOD
005312  104000                        HLT                     ;BDR IS WRONG
005314  012777  110000  173546        MOV     #110000,@CSR    ;GO TO EIR
005322  012777  125252  173542        MOV     #125252,@BDR    ;SET ALT 1'S AND 0'S TO BDR
005330  022777  125252  173534        CMP     #125252,@BDR    ;CHECK PATTERN
005336  001001                        BNE     .+4             ;SHOULD NOT CHECK OUT - OK
005340  104000                        HLT                     ;DATA SHOULD NOT HAVE GOTTEN THROUGH
005342  022777  052525  173522        CMP     #52525,@BDR     ;TEST FOR OLD PATTERN
005350  001401                        BEQ     .+4             ;ALL IS FINE GO TO NEXT TEST
005352  104000                        HLT                     ;BDR IS WRONG
005354  005077  173510                CLR     @CSR            ;GO BACK TO CSR

```

```
1593                                            .SBTTL TEST 41 INCREMENTING PATTERN TO WRAP-AROUND IN BDR
1594                                    ;;*****************************************************************
1595                                    ;       TEST 41 INCREMENTING PATTERN TO WRAP-AROUND IN BDR
1596                                    ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
1597                                    ;;*****************************************************************
1598 005360  000004                     T41:    SCOPE
1599 005362  005077  173502                     CLR     @CSR            ;FORCE TO BE CSR #1
1600 005366  012777  010000  173474             MOV     #10000,@CSR     ;MAINT MODE
1601 005374  005037  014122                     CLR     ICOUNT
1602 005400  005001                             CLR     R1              ;SET-UP
1603 005402  005077  173464                     CLR     @BDR            ;SET-UP
1604 005406  020177  173460             INCDB:  CMP     R1,@BDR         ;SEE IF THEY ARE EQUAL
1605 005412  001401                             BEQ     .+4             ;ARE THEY EQUAL?
1606 005414  104000                             HLT                     ;THEY'RE NOT EQUAL
1607 005416  005277  173450                     INC     @BDR            ;GET NEXT NUMBER
1608 005422  005201                             INC     R1              ;GET NEXT NUMBER
1609 005424  001370                             BNE     INCDB           ;DONE WITH TEST? IF NOT CONTINUE
1610
1611
1612
```

```
                                        .SBTTL TEST 42 TEST THAT GO CLEARS READY AND SETS AGAIN
                                ;;**************************************************************************
                                ;       TEST 42 TEST THAT GO CLEARS READY AND SETS AGAIN
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;;**************************************************************************
005426  000004                          SCOPE
005430  005077  173434                  CLR     @CSR            ;FORCE TO BE CSR #1
005434  012737  002000  014122          MOV     #2000,ICOUNT
005442  012777  010000  173420          MOV     #10000,@CSR     ;MAINT MODE
005450  022777  010200  173412          CMP     #10200,@CSR     ;ONLY READY
005456  001401                          BEQ     .+4
005460  104000                          HLT
005462  012777  177600  173374          MOV     #-200,@WCR      ;SET-UP WCR
005470  013777  001152  173370          MOV     INBUF,@BAR      ;SET-UP BAR
005476  052777  000010  173364          BIS     #10,@CSR        ;FNCT3(NON BURST)
005504  052777  000001  173356          BIS     #1,@CSR         ;GO CLEARS READY
005512  105777  173352                  TSTB    @CSR            ;CHECK READY
005516  100001                          BPL     .+4             ;IS READY CLEAR?
005520  104000                          HLT                     ;READY IS STILL SET
005522  052777  000400  173340          BIS     #400,@CSR       ;SET CYCLE AND START XFERS
005530  005037  001166                  CLR     RDYCHK          ;CLEAR READY CHECK
005534  105777  173330          TSTRDY: TSTB    @CSR            ;CHECK READY
005540  100406                          BMI     DONE            ;IF SET GO TO DONE
005542  062737  000004  001166          ADD     #4,RDYCHK       ;CHECKING TIME FOR READY TO BE SET
005550  100401                          BMI     .+4             ;IF RDYCHK GETS NEGATIVE IT TOOK TOO LONG
005552  000770                          BR      TSTRDY          ;CHECK AGAIN
005554  104000                          HLT                     ;READY GOT CLEARED BUT NEVER SET AGAIN
005556  000240                  DONE:   NOP                     ;GO TO NEXT TEST
```

```
                                        .SBTTL TEST 43 TEST THAT DR11W DOES INTERRUPT WITH PROC AT LEVEL 3
                                ;;*****************************************************************
                                ;       TEST 43 TEST THAT DR11-W DOES INTERRUPT WITH PROC AT LEVEL 3
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;       INTERRUPT VIA ATTN
                                ;;*****************************************************************
005560  000004                  X4:     SCOPE
005562  005077  173302                  CLR     @CSR            ;FORCE TO BE CSR #1
005566  012777  000140  173206          MOV     #140,@PSW       ;STATUS AT LEVEL 3
005574  032777  000200  173266          BIT     #BIT7,@CSR      ;CHECK READY BIT
005602  001010                          BNE     P3INV           ;IS IT SET?
005604  004737  014144                  JSR     PC,CKSWR
005610  000005                          RESET                   ;INIT TO SET READY
005612  032777  000200  173250          BIT     #BIT7,@CSR      ;SEE IF READY IS SET NOW
005620  001001                          BNE     .+4             ;IS IT SET
005622  104000                          HLT                     ;READY CAN'T BE SET BY INIT
005624  012777  005720  173304  P3INV:  MOV     #P3INT,@DRINV   ;SET UP INT VECTOR
005632  012777  000340  173300          MOV     #340,@DRVS
005640  012777  010000  173222          MOV     #10000,@CSR     ;MAINT MODE
005646  052777  000101  173214          BIS     #101,@CSR       ;IE,GO
005654  052777  020004  173206          BIS     #20004,@CSR     ;SET ATTN;ERROR SETS,SETS READY,AND INTERRUPT OCCURS
005662  012737  001000  001312          MOV     #1000,TIME      ;DELAY
005670  005337  001312          22$:    DEC     TIME
005674  001375                          BNE     22$
005676  013777  001140  173232          MOV     DRVS,@DRINV     ;RESTORE INTERRUPT VECTOR
005704  005077  173230                  CLR     @DRVS           ;RESTORE TRAP CATCHER
005710  104000                          HLT                     ;DR11-W DIDN'T INTERRUPT
005712  000005                          RESET                   ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                                ;INTERRUPT REQUEST IN BOARD HDWR.
005714  000137  005740                  JMP     X7
005720  005077  173144          P3INT:  CLR     @CSR            ;CLEAR IE
005724  022626                          CMP     (SP)+,(SP)+     ;REPOSITION THE STACK AFTER AN INTERRUPT
005726  013777  001140  173202          MOV     DRVS,@DRINV     ;RESTORE INTERRUPT VECTOR
005734  005077  173200                  CLR     @DRVS           ;RESTORE TRAP CATCHER
```

```
                                        .SBTTL TEST 44 THAT DR11W DOES NOT INTERRUPT WITH PROC AT LEVEL 7
                                ;;******************************************************************
                                ;       TEST 44 TEST THAT DR11-W DOES NOT INTERRUPT WITH PROC AT LEVEL 7
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;       INTERRUPT VIA ATTN
                                ;;******************************************************************
005740  000004                  X7:     SCOPE
005742  005077  173122                  CLR     @CSR            ;FORCE TO BE CSR #1
005746  005037  014122                  CLR     ICOUNT
005752  012777  000340  173022          MOV     #340,@PSW       ;STATUS AT LEVEL 7
005760  032777  000200  173102          BIT     #BIT7,@CSR      ;CHECK READY BIT
005766  001010                          BNE     P7INV           ;IS IT SET
005770  004737  014144                  JSR     PC,CKSWR
005774  000005                          RESET                   ;INIT TO SET READY
005776  032777  000200  173C64          BIT     #BIT7,@CSR      ;SEE IF READY IS SET NOW
006004  001001                          BNE     .+4             ;IS READY SET?
006006  104000                          HLT                     ;READY CAN'T BE SET BY INIT
006010  012777  006100  173120  P7INV:  MOV     #P7ERR,@DRINV   ;SET UP INT VECTOR
006016  012777  000340  173114          MOV     #340,@DRVS
006024  012777  010000  173036          MOV     #10000,@CSR     ;MAINT MODE
006032  052777  000101  173030          BIS     #101,@CSR       ;IE,GO
006040  052777  020004  173022          BIS     #20004,@CSR     ;SET ATTN;ERROR SETS,SETS READY,AND INTERRUPT OCCURS
006046  012737  001000  001312          MOV     #1000,TIME      ;DELAY
006054  005337  001312          22$:    DEC     TIME
006060  001375                          BNE     22$
006062  000005                          RESET                   ;CLR IE,OUTSTANDING INTERR RFQUEST
006064  013777  001140  173044          MOV     DRVS,@DRINV
006072  005077  173042                  CLR     @DRVS
006076  000411                          BR      X1
006100  022626                  P7ERR:  CMP     (SP)+,(SP)+     ;RESTORE STACK
006102  005077  172762                  CLR     @CSR            ;CLEAR IE
006106  013777  001140  173022          MOV     DRVS,@DRINV
006114  005077  173020                  CLR     @DRVS
006120  104000                          HLT                     ;DR11-W INTERRUPTED, BUT IT SHOULDN'T HAVE
```

```
                                        .SBTTL TEST 45 THAT DR11W DOES INTERRUPT AND INDICATE BR LEVEL
                                ;;******************************************************************
                                ;       TEST 45 THAT DR11W DOES INTERRUPT AND INDICATE BR LEVEL
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;       INTERRUPT VIA ATTN
                                ;;******************************************************************
006122  000004                  X1:     SCOPE
006124  012777  000340  172650          MOV     #340,@PSW       ;STATUS AT LEVEL 7
006132  005077  172732          NSTAT:  CLR     @CSR            ;FORCE TO BE CSR #1
006136  032777  000200  172724          BIT     #BIT7,@CSR      ;CHECK READY BIT
006144  001010                          BNE     PINV            ;IS IT SET
006146  004737  014144                  JSR     PC,CKSWR
006152  000005                          RESET                   ;INIT TO SET READY
006154  032777  000200  172706          BIT     #BIT7,@CSR      ;SEE IF READY IS SET NOW
006162  001001                          BNE     .+4             ;IS READY SET?
006164  104000                          HLT                     ;READY CAN'T BE SET BY INIT
006166  012777  006272  172742  PINV:   MOV     #BRLEV,@DRINV   ;SET UP INT VECTOR
006174  012777  000340  172736          MOV     #340,@DRVS
006202  012777  010000  172660          MOV     #10000,@CSR     ;MAINT MODE
006210  052777  000101  172652          BIS     #101,@CSR       ;IE,GO
006216  052777  020004  172644          BIS     #20004,@CSR     ;SET ATTN;ERROR SETS,SETS READY,AND INTERRUPT OCCURS
006224  012737  001000  001312          MOV     #1000,TIME      ;DELAY
006232  005337  001312          22$:    DEC     TIME
006236  001375                          BNE     22$
006240  000005                          RESET                   ;CLR IE,OUTSTANDING INTERR REQUEST
006242  013777  001140  172666          MOV     DRVS,@DRINV
006250  005077  172664                  CLR     @DRVS
006254  017737  172522  023412          MOV     @PSW,LEVEL      ;SAVE OLD STATUS LEVEL
006262  162777  000040  172512          SUB     #40,@PSW        ;NEW PSW
006270  000720                          BR      NSTAT           ;TRY AGAIN
006272  022626                  BRLEV:  CMP     (SP)+,(SP)+     ;RESTORE STACK
006274  005077  172570                  CLR     @CSR            ;CLEAR IE
006300  013777  001140  172630          MOV     DRVS,@DRINV
006306  005077  172626                  CLR     @DRVS
006312  042737  177437  023412          BIC     #177437,LEVEL   ;GET
006320  013701  023412                  MOV     LEVEL,R1        ;BR
006324  006201                          ASR     R1              ;INTERRUPT
006326  006201                          ASR     R1              ;LEVEL
006330  006201                          ASR     R1
006332  006201                          ASR     R1
006334  006201                          ASR     R1
006336  010137  023412                  MOV     R1,LEVEL
006342  104401  001351                  TYPE    ,$CRLF
006346  104401  001351                  TYPE    ,$CRLF
006352  104401  006360                  TYPE    ,65$            ;;TYPE ASCIZ STRING
006356  000421                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  /TEST 40: DR11W INTERRUPT LEVEL =/
006422                          64$:
006422  013746  023412                  MOV     LEVEL,-(SP)     ;;SAVE LEVEL FOR TYPEOUT
006426  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
006430  104401  001351                  TYPE    ,$CRLF
006434  104401  001351                  TYPE    ,$CRLF
```

```
                                        .SBTTL TEST 46 THAT GIVING A GO WITHOUT CLEARING ERROR CAUSES INTRPT
                                ;;**************************************************************************
                                ;       TEST 46 TEST THAT GIVING A GO WITHOUT CLEARING A PREVIOUS
                                ;       ERROR CAUSES ANOTHER INTERRUPT
                                ;       FOR THIS TEST, THE NORMAL INTERRUPT MECHANISM SHOULD BE WORKING
                                ;       MAINTENANCE MODE(INTERNAL WRAP-AROUND)
                                ;;**************************************************************************
006440  000004                  T55:    SCOPE
006442  005077  172422                  CLR     @CSR            ;FORCE TO BE CSR #1
006446  012777  006534  172462          MOV     #ERRD0,@DRINV   ;INTERRUPT VECTOR TO ERRD0
006454  012777  000140  172456          MOV     #140,@DRVS      ;INTERRUPT STATUS TO LEVEL 4
006462  005077  172314                  CLR     @PSW            ;LET THE DR11-W INTERRUPT
006466  012777  010000  172374          MOV     #10000,@CSR     ;MAINT MODE
006474  052777  000101  172366          BIS     #101,@CSR       ;IE ,GO
006502  052777  020004  172360          BIS     #20004,@CSR     ; SET ATTN AND INTERRUPT
006510  012737  001000  001312          MOV     #1000,TIME      ;DELAY
006516  005337  001312          22$:    DEC     TIME
006522  001375                          BNE     22$
006524  104000                          HLT                     ;NO DR11-W INTERRUPT
006526  000005                          RESET                   ;CLEAR OUTSTANDING INTERR. REQST.
006530  000137  006632                  JMP     T60
006534  022626                  ERRD0:  CMP     (SP)+,(SP)+     ;READJUST STACK
006536  005777  172326                  TST     @CSR            ;TEST CSR
006542  100401                          BMI     .+4             ;ERROR SET?
006544  104000                          HLT                     ;ERROR IS CLEAR - SHOULD HAVE ERROR
006546  012777  006614  172362          MOV     #ERRD01,@DRINV  ;INTERRUPT VECTOR TO ERRD01
006554  005077  172306                  CLR     @BAR            ;PREVENT CAUSING ANOTHER ERROR
006560  012777  177777  172276          MOV     #-1,@WCR        ;SET-UP WCR
006566  005277  172276                  INC     @CSR            ;GO TO CSR
006572  005037  006600                  CLR     1$+2
006576  005227  000001          1$:     INC     #1              ;DELAY;WAIT FOR INTERRUPT
006602  001375                          BNE     1$
006604  104000                          HLT                     ;NO DR11-W INTERRUPT
006606  000005                          RESET                   ;CLEAR INTERRUPT REQUEST
006610  000137  006632                  JMP     T60
006614  022626                  ERRD01: CMP     (SP)+,(SP)+
006616  005777  172246                  TST     @CSR            ;CHECK ERROR
006622  100001                          BPL     .+4             ;ERROR SET?
006624  104000                          HLT                     ;ERROR IS SET - SHOULD BE CLEAR
                                                                ;PREVIOUS ERROR WAS CLEAR
006626  004737  010156                  JSR     PC,NORMAL
```

```
                                        .SBTTL TEST 47 THAT FUNCTION BITS INCREMENT WITH MAINT MODE TRANSFERS
                                ;;*****************************************************************************
                                ;       TEST 47 TEST THAT FUNCTION BITS INCREMENT WITH MAINT MODE TRANSFERS
                                ;;*****************************************************************************
006632  000004                T60:    SCOPE
006634  005077  172230                CLR     @CSR                    ;FORCE TO BE CSR
006640  012737  002000  014122        MOV     #2000,ICOUNT
006646  013777  001152  172212        MOV     INBUF,@BAR              ;SET-UP BAR
006654  012737  000001  001202        MOV     #1,FNCCNT               ;GET READY FOR CHECKING
006662  012777  006762  172246 MFLOOP: MOV     #T60CHK,@DRINV          ;INTERRUPT VECTOR
006670  013777  001074  172242        MOV     DRINL,@DRVS             ;INTERRUPT VECTOR PRIORITY TO DRINL
006676  013737  001202  001306        MOV     FNCCNT,TEMP             ;NUMBER OF TRANSFERS
006704  005437  001306                NEG     TEMP                    ;SET UP FOR WCR
006710  013777  001306  172146        MOV     TEMP,@WCR               ;SET-UP FOR 1 TRANSFER
006716  005077  172060                CLR     @PSW                    ;LET THE DR11-W INTERRUPT
006722  012777  010000  172140        MOV     #10000,@CSR             ;MAINT MODE
006730  052777  000501  172132        BIS     #501,@CSR               ;IE,CYCLE & GO
006736  012737  001000  00 312        MOV     #1000,TIME              ;WAIT FOR INTERRUPT
006744  005337  001312          1$:   DEC     TIME
006750  001375                        BNE     1$
006752  104000                        HLT                             ;NO INTERRUPT OCCURRED
006754  000005                        RESET                           ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                                        ;INTERRUPT REQUEST IN BOARD HDWR.
006756  000137  007024                JMP     T61                     ;GO TO NEXT TEST
006762  022626                T60CHK: CMP     (SP)+,(SP)+
006764  117701  172100                MOVB    @CSR,R1                 ;LOWER BYTE OF CSR TO R1
006770  042701  000700                BIC     #700,R1                 ;GET RID OF READY,CYCLE,IF BECAUSE OF MAINT MODE
006774  006201                        ASR     R1                      ;MOVE IT RIGHT ONE PLACE
006776  123701  001202                CMPB    FNCCNT,R1               ;CHECK AGAINST FNCCNT
007002  001401                        BEQ     .+4                     ;SHOULD BE EQUAL
007004  104000                        HLT                             ;FUNCTION BITS DIDN'T INCREMENT IN MAINT MODE
007006  005237  001202                INC     FNCCNT                  ;GET READY FOR NEXT PASS
007012  022737  000010  001202        CMP     #10,FNCCNT              ;ONLY 10 BECAUSE FNCT3-1 GO TO ZERO
007020  001401                        BEQ     T61                     ;IF ITS EQUAL GO CHECK DATA
007022  000717                        BR      MFLOOP                  ;DO IT AGAIN
```

```
                                        .SBTTL TEST 50 TEST FOR 10 MAINT MODE TRANSFERS
                                ;;*****************************************************************
                                ;       TEST 50 TEST FOR 10 MAINT MODE TRANSFERS
                                ;;*****************************************************************
007024  000004                  T61:    SCOPE
007026  005077  172036                  CLR     @CSR            ;FORCE TO BE CSR #1
007032  012737  000010  001156          MOV     #10,BUFLEN      ;BUFLEN=10
007040  013737  001156  001164          MOV     BUFLEN,WCLEN    ;PREPARE NUMBER FOR WCR
007046  005437  001164                  NEG     WCLEN           ;2'S COMPLEMENT OF BUFLEN
007052  004737  007714                  JSR     PC,LODBUF       ;LOAD IN BUFFER WITH INCREMENTING PATTERN
007056  004737  007752                  JSR     PC,CHKBFF       ;LOAD CHECK BUFFER WITH MODIFIED INCREMENTING PATTERN
007062  013777  001164  171774          MOV     WCLEN,@WCR      ;SET UP WCR
007070  013777  001152  171770          MOV     INBUF,@BAR      ;SET UP BAR
007076  012777  177777  171766          MOV     #-1,@BDR        ;MAINT AIDE
007104  012777  007174  172024          MOV     #T61CHK,@DRINV  ;INTERRUPT VECTOR
007112  013777  001074  172020          MOV     DRINL,@DRVS     ;INTERRUPT STATUS AT PRIORITY DRINL
007120  005077  171656                  CLR     @PSW            ;LET DR11-W INTERRUPT
007124  012777  010000  171736          MOV     #10000,@CSR     ;MAINT MODE
007132  052777  000501  171730          BIS     #501,@CSR       ;IE,CYCLE & GO
007140  012737  001000  001312          MOV     #1000,TIME      ;WAIT FOR INTERRJPT
007146  005337  001312          1$:     DEC     TIME
007152  001375                          BNE     1$
007154  104000                          HLT                     ;NO INTERRUPT OCCURED
007156  000005                          RESET                   ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                                ;INTERRUPT REQUEST IN BOARD HDWR.
007160  004737  010016                  JSR     PC,INTA
007164  004737  010116                  JSR     PC,DATCHK
007170  000137  007206                  JMP     T61.5
007174  022626                  T61CHK: CMP     (SP)+,(SP)+
007176  004737  010016                  JSR     PC,INTA         ;CLEAR IE,CHECK ERROR,READY,WC 0,& BAR
007202  004737  010116                  JSR     PC,DATCHK
```

```
1882                                            .SBTTL TEST 51 10 MAINTENANCE MODE XFERS
1883                                    ;;******************************************************************
1884                                    ;       TEST 51 TEST THAT DOING 10 MAINTENANCE MODE XFERS
1885                                    ;       WITHOUT EXPERIENCING AN INTERRUPT,WILL INHIBIT
1886                                    ;       'GO' FROM INITIATING FURTHER XFERS
1887                                    ;;******************************************************************
1888 007206  000004                    T61.5:  SCOPE
1889 007210  005077  171654                    CLR     @CSR            ;FORCE TO BE CSR #1
1890 007214  012737  000010  014122            MOV     #10,ICOUNT
1891 007222  012777  000340  171552            MOV     #340,@PSW       ;NO CPU INTERRUPT
1892 007230  012737  001000  001312            MOV     #1000,TIME      ;SET DELAY
1893 007236  012737  000010  001156            MOV     #10,BUFLEN      ;BUFLEN=10
1894 007244  013737  001156  001164            MOV     BUFLEN,WCLEN    ;PREPARE NUMBER FOR WCR
1895 007252  005437  001164                    NEG     WCLEN           ;2'S COMPLEMENT OF BUFLEN
1896 007256  004737  007714                    JSR     PC,LODBUF       ;LOAD IN BUFFER WITH INCREMENTING PATTERN
1897 007262  004737  007752                    JSR     PC,CHKBFF       ;LOAD CHECK BUFFER WITH MODIFIED INCREMENTING PATTERN
1898 007266  013777  001164  171570            MOV     WCLEN,@WCR      ;SET UP WCR
1899 007274  013777  001152  171564            MOV     INBUF,@BAR      ;SET UP BAR
1900 007302  012777  177777  171562            MOV     #-1,@BDR        ;MAINT AIDE
1901 007310  012777  007360  171620            MOV     #WRONG1,@DRINV  ;INTERRUPT VECTOR
1902 007316  013777  001074  171614            MOV     DRINL,@DRVS     ;INTERRUPT STATUS AT PRIORITY DRINL
1903 007324  012777  010000  171536            MOV     #10000,@CSR     ;MAINT MODE
1904 007332  052777  000501  171530            BIS     #501,@CSR       ;IE,CYCLE & GO
1905 007340  005337  001312          1$:       DEC     TIME            ;WAIT FOR XFERS COMPLETE
1906 007344  001375                            BNE     1$
1907 007346  004737  010016                    JSR     PC,INTA         ;CLR IE,CHECK ERROR,READY,WC=0,BAR
1908 007352  004737  010116                    JSR     PC,DATCHK       ;CHECK PROPER DATA
1909 007356  000404                            BR      OK
1910 007360  022626                    WRONG1: CMP     (SP)+,(SP)+     ;SHOULD NOT INTERRUPT
1911 007362  104000                            HLT                     ;NO INTERRUPT 1ST TIME
1912 007364  000137  007514                    JMP     NRM
1913 007370  012777  010000  171472    OK:     MOV     #10000,@CSR
1914 007376  012737  001000  001312            MOV     #1000,TIME
1915 007404  013777  001164  171452            MOV     WCLEN,@WCR
1916 007412  013777  001152  171446            MOV     INBUF,@BAR
1917 007420  012777  007510  171510            MOV     #WRONG2,@DRINV
1918 007426  012777  010000  171434            MOV     #10000,@CSR     ;MAINT MODE
1919 007434  052777  000501  171426            BIS     #501,@CSR       ;IE,CYCLE,GO
1920 007442  005337  001312          3$:       DEC     TIME
1921 007446  001375                            BNE     3$
1922 007450  022777  010700  171412            CMP     #10700,@CSR     ;ONLY READY,MAINT,IE,& CYCLE
1923 007456  001401                            BEQ     .+4
1924 007460  104000                            HLT                     ;CSR IN ERROR
1925 007462  022777  177770  171374            CMP     #-10,@WCR       ;CHECK THAT NO XFERS WERE MADE
1926 007470  001004                            BNE     5$
1927 007472  023777  001152  171366            CMP     INBUF,@BAR      ;
1928 007500  001401                            BEQ     RSET
1929 007502  104000                    5$:     HLT                     ;XFERS SHOULD HAVE BEEN INHIBITED
1930 007504  000005                    RSET:   RESET                   ;WILL ELIMINATE ANY OUTSTANDING INTERRUPT REQUESTS
1931 007506  000402                            BR      NRM
1932 007510  022626                    WRONG2: CMP     (SP)+,(SP)+     ;READJUST STACK
1933 007512  104000                            HLT                     ;SHOULD NOT INTERRUPT 2ND TIME
1934 007514  004737  010156            NRM:    JSR     PC,NORMAL
1935
```

```
                                        .SBTTL TEST 52 TEST FOR 200 NPR TRANSFERS IN MAINT MODE
                                ;;**************************************************************************
                                ;       TEST 52 TEST FOR 200 NPR TRANSFERS IN MAINT MODE
                                ;;**************************************************************************
007520  000004                          SCOPE
007522  005077  171342                  CLR     @CSR            ;FORCE TO BE CSR #1
007526  012737  002000  014122          MOV     #2000,ICOUNT
007534  012737  000200  001156          MOV     #200,BUFLEN     ;LENGTH OF BUFFER = 200
007542  013737  001156  001164          MOV     BUFLEN,WCLEN    ;PREPARE NUMBER FOR WCR
007550  005437  001164                  NEG     WCLEN           ;2'S COMPLEMENT OF BUFLEN
007554  004737  007714                  JSR     PC,LODBUF       ;LOAD INBUF WITH INCREMENTING PATTERN
007560  004737  007752                  JSR     PC,CHKBFF       ;LOAD CHKBUF WITH MODIFIED INCREMENTED PATTERN
007564  013777  001164  171272          MOV     WCLEN,@WCR      ;SET UP WCR
007572  013777  001152  171266          MOV     INBUF,@BAR      ;SET UP BAR
007600  012777  000001  171264          MOV     #1,@BDR         ;MAINT AIDE
007606  012777  007676  171322          MOV     #T62CHK,@DRINV  ;INT VECTOR
007614  013777  001074  171316          MOV     DRINL,@DRVS     ;INT VECTOR AT PRIORITY DRINL
007622  005077  171154                  CLR     @PSW            ;LET THE DR11-W INTERRUPT
007626  012777  010000  171234          MOV     #10000,@CSR     ;MAINT MODE
007634  052777  000501  171226          BIS     #501,@CSR       ;IE,CYCLE & GO
007642  012737  001000  001312          MOV     #1000,TIME
007650  005337  001312          1$:     DEC     TIME
007654  001375                  BNE     1$
007656  104000                  HLT                             ;NO INTERRUPT OCCURED
007660  000005                  RESET                           ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                                ;INTERRUPT REQUEST IN BOARD HDWR.
007662  004737  010016                  JSR     PC,INTA
007666  004737  010116                  JSR     PC,DATCHK
007672  000137  010276                  JMP     T42
007676  022626          T62CHK: CMP     (SP)+,(SP)+
007700  004737  010016                  JSR     PC,INTA         ;CLR IF,CHECK ERROR,RESDY,WC 0,& BAR
007704  004737  010116                  JSR     PC,DATCHK       ;CHECK THAT CORRECT DATA WAS TRANSFERRED
007710  000137  010276                  JMP     T42
```

```
                                LODBUF: MOV     INBUF,R2        ;MOVE STARTING ADDRESS OF INBUF TO R2
                                        CLR     LENCHK          ;CLEAR LENGTH CHECK
                                        CLR     (R2)+           ;CLEAR STARTING ADDRESS OF INBUFF AND INC BY 2
                                LOADA:  INC     LENCHK          ;INC LENGTH CHECK BY 1
                                        CMP     LENCHK,BUFLEN   ;CHECK FOR DONE
                                        BEQ     LDEXIT          ;IS INBUF FILLED?
                                        MOV     LENCHK,(R2)+    ;LOAD NEXT BUFFER WORD
                                        BR      LOADA           ;CONTINUE CHECKING
                                LDEXIT: RTS     PC              ;EXIT
                                CHKBFF: MOV     CHKBUF,R2       ;STARTING ADDRESS OF CHECK-BUFFER TO R2
                                        CLR     LENCHK          ;CLEAR LENGTH CHECK
                                        CLR     R3              ;CLEAR R3
                                CHKA:   MOV     R3,(R2)+        ;MOVE R3 TO CHKBUF ADDRESS AND INC BY 2
                                        MOV     R3,(R2)+        ;MOVE R3 TO NEXT CHKBUF ADDRESS AND INC BY 2
                                        ADD     #2,LENCHK       ;ADD 2 TO LENGTH CHECK
                                        CMP     LENCHK,BUFLEN   ;CHECK FOR DONE
                                        BPL     .+10            ;IS CHECK-BUFFER FILLED?
                                        ADD     #2,R3           ;NEXT NUMBER FOR BUFFER
                                        BR      CHKA            ;CONTINUE FILLING
                                        RTS     PC              ;EXIT
                                INTA:   BIC     #BIT6,@CSR      ;CLEAR IE
                                        TST     @CSR            ;CHECKING FOR ERROR
                                        BPL     .+4             ;ERROR SET?
                                        HLT                     ;ERROR BIT IS SET
                                        TSTB    @CSR            ;CHECKING READY BIT
                                        BMI     .+4             ;IS READY SET
                                        HLT                     ;FALSE INTERRUPT - ERROR AND READY ARE CLEAR
                                        TST     @WCR            ;TEST1 FOR WCR=0
                                        BEQ     .+4             ;WAS IT EQUAL?
                                        HLT                     ;WC NOT = 0
                                        MOV     BUFLEN,R2       ;BUFFER LENGTH TO R2
                                        ADD     BUFLEN,R2       ;NUMBER OF XFERS TIMES 2
                                        ADD     INBUF,R2        ;CORRECT BAR
                                        TST     CABLE           ;IS THIS CABLE TESTING?
                                        BEQ     1$              ;NO
                                        INC     R2              ;CABLE MODE TESTING LEAVES BIT 0
                                                                ;OF BAR SET.THEREFORE MUST CHECK
                                                                ;FOR ODD ADDRESS.
                                1$:     CMP     @BAR,R2         ;CHECKKING BAR
                                        BEQ     .+4             ;IS BAR CORRECT
                                        HLT                     ;NO
                                        JSR     PC,NORMAL
                                        RTS     PC              ;EXIT

                                DATCHK: MOV     CHKBUF,%2       ;STARTING ADDRESS OF CHECK BUFFER TO R2
                                        MOV     INBUF,%3        ;STARTING ADDRESS OF IN BUFFER TO R3
                                        CLR     LENCHK          ;CLEAR LENGTH CHECK
                                COMPAR: INC     LENCHK          ;MAKE A COMPARISON
                                        CMP     (%2)+,(%3)+     ;IS THE DATA CORRECT?
                                        BEQ     .+4             ;BRANCH IF OK
                                        HLT                     ;BAD DATA
                                        CMP     LENCHK,BUFLEN   ;SEE IF THE BUFFER HAS BEEN CHECKED
                                        BNE     COMPAR          ;BUFFER CHECKED?
                                        RTS     %7
                                NORMAL: MOV     DRVS,@DRINV     ;RESTORE DR11-B INTERRUPT VECTOR
                                        CLR     @DRVS           ;RESTORE DR11-B INTERRUPT STATUS
                                        MOV     #340,@PSW       ;RESTORE PROC TO PRIORITY LEVEL 7
```

```
2028 010176  000207                          RTS     %7              ;EXIT
2029 010200  012702  052525          DATOCK: MOV     #52525,%2       ;DATO NUMBER TO R2
2030 010204  013703  001152                  MOV     INBUF,%3        ;STARTING ADDRESS OF IN BUFFER TO R3
2031 010210  005037  001160                  CLR     LENCHK          ;CLEAR LENGTH CHECK
2032 010214  005237  001160          COMPRR: INC     LENCHK          ;MAKE A COMPARISON
2033 010220  020223                          CMP     %2,(%3)+        ;IS THE DATA CORRECT?
2034 010222  001401                          BEQ     .+4             ;BRANCH IF OK
2035 010224  104000                          HLT                     ;BAD DATA
2036 010226  023737  001160  001156          CMP     LENCHK,BUFLEN   ;SEE IF THE BUFFER HAS BEEN CHECKED
2037 010234  001367                          BNE     COMPRR          ;BUFFER CHECKED?
2038 010236  020223                          CMP     %2,(%3)+        ;CHECK END OF BUFFER + 1
2039 010240  001001                          BNE     .+4             ;SEE IF TOO MANY WORDS WERE TRANSFERRED
2040 010242  104000                          HLT                     ;TOO MANY
2041 010244  000207                          RTS     %7              ;EXIT
2042 010246  042777  000100  170614  ERRCHK: BIC     #BIT6,@CSR      ;CLEAR IE
2043 010254  005777  170610                  TST     @CSR            ;CHECKING FOR ERROR
2044 010260  100001                          BPL     .+4             ;ERROR SET?
2045 010262  104000                          HLT                     ;ERROR BIT IS SET
2046 010264  105777  170600                  TSTB    @CSR            ;CHECKING READY BIT
2047 010270  100401                          BMI     .+4             ;IS RDY SET
2048 010272  104000                          HLT                     ;FALSE ENTRY - ERROR AND READY ARE CLEAR
2049 010274  000207                          RTS     %7              ;EXIT
2050
2051
```

```
                                        .SBTTL TEST 53 THAT DOING A DATO TO THE DIODE MEMORY CAUSES NEX
                                ;;*******************************************************************
                                ;       TEST 53 TEST THAT DOING A DATO TO THE DIODE MEMORY CAUSES NEX
                                ;       MAINTENANCE MODE(INTERNAL WRAP AROUND)
                                ;;*******************************************************************
010276  000004                T42:    SCOPE
010300  005077  170564                CLR     @CSR            ;FORCE TO BE CSR #1
010304  012777  177776  170552        MOV     #-2,@WCR        ;SET UP WCR
010312  013777  001150  170546        MOV     DIOMEM,@BAR     ;SET UP BAR
010320  012777  010406  170610        MOV     #NEXCHK,@DRINV  ;INTERRUPT VECTOR TO NEXCHK
010326  013777  001074  170604        MOV     DRINL,@DRVS     ;INTERRUPT STATUS TO LEVEL DRINL
010334  005077  170442                CLR     @PSW            ;LET THE DR11-W INTERRUPT
010340  012777  010000  170522        MOV     #10000,@CSR     ;MAINT MODE
010346  052777  000062  170514        BIS     #62,@CSR        ;FNCT1,XBA16,XBA17
010354  052777  000501  170506        BIS     #501,@CSR       ;IE,CYCLE,GO
010362  012737  001000  001312        MOV     #1000,TIME      ;DELAY
010370  005337  001312        22$:    DEC     TIME
010374  001375                        BNE     22$
010376  104000                        HLT                     ;NO DR11-W INTERRUPT
010400  000005                        RESET                   ;CLEAR ANY OUTSTANDING INTERRUPT REQUESTS
010402  000137  010450                JMP     SCOP
010406  017705  170456        NEXCHK: MOV     @CSR,R5         ;SAVE CSR IN R5
010412  005705                        TST     R5              ;TEST CSR
010414  100401                        BMI     .+4             ;ERROR SET?
010416  104000                        HLT                     ;ERROR NOT SET
010420  105705                        TSTB    R5              ;TEST FOR READY
010422  001001                        BNE     .+4             ;READY SET?
010424  104000                        HLT                     ;READY ISN'T SET
010426  032705  040000        10$:    BIT     #BIT14,R5       ;CHECK NEX
010432  001001                        BNE     .+4             ;NEX SET?
010434  104000                        HLT                     ;NEX IS CLEAR
010436  005077  170426                CLR     @CSR            ;RETURN TO CSR #1
010442  022626                        CMP     (SP)+,(SP)+     ;RESTORE THE STACK
010444  004737  010156                JSR     PC,NORMAL
010450  000004                SCOP:   SCOPE
```

```
                                        .SBTTL TEST 54 CROSSING A 32K BOUNDRY DOESNT CAUSE A BAOF OR FORCE ERROR
                                ;;**************************************************************************
                                ;       TEST 54 TEST THAT CROSSING A 32K BOUNDRY DOES NOT CAUSE
                                ;       A BAOF AND DOES NOT FORCE AN ERROR
                                ;;**************************************************************************
010452  000004                        SCOPE
010454  005077  170410                CLR     @CSR                    ;FORCE TO BE CSR #'
010460  012737  000010  014122        MOV     #10,ICOUNT
010466  012777  177760  170370        MOV     #-20,@WCR               ;SET UP WCR
010474  012777  177776  170364        MOV     #-2,@BAR                ;SET UP BAR FOR PROC STATUS ADDRESS
010502  012777  010546  170426        MOV     #BAOFCK,@DRINV          ;INTERRUPT VECTOR TO BAOFCK
010510  013777  001074  170422        MOV     DRINL,@DRVS             ;INTERRUPT STATUS TO LEVEL DRINL
010516  012737  001000  001312        MOV     #1000,TIMF
010524  005077  170252                CLR     @PSW                    ;LET THE DR11-W INTERRUPT
010530  012777  000563  170332        MOV     #563,@CSR               ;CYCLE,IE, FNCT1, XBA17, XBA16, AND GO TO CSR
010536  005337  001312        1$:     DEC     TIME
010542  001375                        BNE     1$                      ;WAIT FOR INTERRUPT
010544  104000                        HLT                             ;NO INTERRUPT
010546  017705  170316        BAOFCK: MOV     @CSR,R5                 ; SAVE CSR
010552  022626                        CMP     (SP)+,(SP)+             ;RESTORE THE STACK
010554  005705                        TST     R5                      ;TEST CSR
010556  100401                        BMI     .+4                     ;ERROR SET?
010560  104000                        HLT                             ;ERROR NOT SET
010562  105705                        TSTB    R5                      ;TEST FOR READY
010564  100401                        BMI     .+4                     ;READY SET?
010566  104000                        HLT                             ;READY ISN'T SET
010570  032705  040000                BIT     #40000,R5               ;CHECK NEX
010574  001001                        BNE     .+4                     ;ARE THEY CLEAR?
010576  104000                        HLT                             ;NEX IS CLEAR
010600  005077  170262                CLR     @BAR                    ;CLEAR BUS ADDRESS REGISTER
010604  005077  170260                CLR     @CSR                    ;RETURN TO CSR #1
010610  004737  014144                JSR     PC,CKSWR
010614  000005                        RESET                           ;INIT
010616  004737  010156                JSR     PC,NORMAL


                                ;THIS ENDS MAINTENANCE MODE TESTING
```

```
                                .SBTTL
                                .SBTTL          CABLE MODE TESTING
                                .SBTTL
                                                ;CABLE MODE TESTING(WRAP-AROUND
                                                        ;CABLE IN USER SLOTS)

                                        ;TESTS 55 THRU 62 ARE PERFORMED IF:
                                ;       1.SWITCH REGISTER 10 IS SET,OR
                                ;       2.SWITCH REG. 10 & 9 ARE DOWN(0),AND THE CABLE IS
                                ;       IN PLACE:THE PROGRAM WILL TEST FOR CABLE.

                                ;THE FOLLOWING ROUTINE DETERMINES IF CABLE IS IN
                                ;OR SWITCH REGISTER OPTION IS SELECTED
010622  005077  170242          CBL:    CLR     @CSR            ;FORCE TO BE CSR #1
010626  004737  014144                  JSR     PC,CKSWR
010632  032777  002000  170466          BIT     #2000,@SWR      ;MAINTENANCE MODE TESTS ONLY?
010640  001402                          BEQ     1$              ;NO
010642  000137  013132                  JMP     END             ;YES
010646  032777  001000  170452  1$:     BIT     #1000,@SWR      ;PERFORM CABLE TESTS UNCONDITIONALLY?
010654  001402                          BEQ     2$              ;NO;DETERMINE CABLE STATUS BY PROGRAM DETECTION
010656  000137  010722                  JMP     YESCBL
010662  000005                  2$:     RESET
010664  005077  170200                  CLR     @CSR            ;FORCE CSR #1
010670  017701  170174                  MOV     @CSR,R1         ;CHECK CSR
010674  032701  127000                  BIT     #127000,R1      ;TEST FOR ERROR,ATTN,DSTATA,DSTATB,DSTATC
010700  001015                          BNE     NOCBL
010702  112777  000004  170160          MOVB    #4,@CSR         ;IF ATTN IS SET,CABLE IS IN
010710  017701  170154                  MOV     @CSR,R1
010714  032701  020000                  BIT     #20000,R1
010720  001405                          BEQ     NOCBL
010722  012737  000001  001314  YESCBL: MOV     #1,CABLE
010730  000137  010744                  JMP     T33
010734  005037  001314          NOCBL:  CLR     CABLE           ;DON'T DO CABLE TESTS
010740  000137  013132                  JMP     END
```

```
                                        .SBTTL TEST 55 TEST FOR 1 DATI NON BURST MODE TRANSFER
                                ;;*****************************************************************
                                ;       TEST 55 TEST FOR 1 DATI NON BURST MODE  TRANSFER
                                ;       CABLE MODE(WITH WRAP AROUND CABLE IN USER SLOTS)
                                ;;*****************************************************************
010744  005077  170120          T33:    CLR     @CSR            ;FORCE TO BE CSR #1
010750  012737  010764  014126          MOV     #T33STR,RETURN  ;REVISE SCOPE RETURN
010756  012737  000001  001314          MOV     #1,CABLE        ;PRINT CABLE MESGE WITH ERROR PRINT
010764                          T33STR:
010764  005077  170100                  CLR     @CSR
010770  005777  170074          TNPR1:  TST     @CSR            ;CHECK ERROR BIT
010774  100001                          BPL     .+4             ;IS IT CLEAR?
010776  104000                          HLT                     ;ERROR SHOULD NOT BE SET
011000  022777  000200  170062          CMP     #200,@CSR       ;CHECK READY
011006  001401                          BEQ     .+4             ;
011010  104000                          HLT                     ;SOMETHING OTHER THAN READY IS SET
011012  012777  177777  170044  NPRRDY: MOV     #-1,@WCR        ;SET UP FOR 1 TRANSFER
011020  012777  001146  170040          MOV     #NPR1,@BAR      ;TRANSFER FROM BUS ADDRESS IN NPR1
011026  005077  170040                  CLR     @BDR            ;GET READY TO RECEIVE DATA
011032  012737  052525  001146          MOV     #52525,NPR1     ;SET UP TRANSFER DATA
011040  012777  011114  170070          MOV     #INTB,@DRINV    ;INTERRUPT VECTOR TO INTB
011046  012777  000005  170064          MOV     #5,@DRVS        ;INTERRUPT PRIORITY TO LEVEL 5
011054  005077  167722                  CLR     @PSW            ;LET THE DR11-W INTERRUPT
011060  012777  000110  170002          MOV     #110,@CSR       ;NON BURST(FNCT3),IE
011066  052777  000401  167774          BIS     #401,@CSR       ;CYCLE,GO
011074  005037  011102                  CLR     1$+2            ;WAIT FOR NPR AND INTERRUPT
011100  005227  000001          1$:     INC     #1
011104  001375                          BNE     1$
011106  104000                          HLT                     ;NO DR11-W INTERRUPT
011110  000005                          RESET                   ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                                ;INTERRUPT REQUEST IN BOARD HDWR.
011112  000424                          BR      T33CLR          ;CLEAR IE
011114  004737  010246          INTB:   JSR     PC,ERPCHK       ;CLEAR IE,CHECK FOR ERROR
011120  005777  167740                  TST     @WCR            ;TEST WCR
011124  001401                          BEQ     .+4             ;IS WCR EQUAL TO ZERO?
011126  104000                          HLT                     ;WCR NOT EQUAL TO ZERO
011130  022777  001151  167730          CMP     #NPR1+3,@BAR    ;COMPARE CORRECT BAR WITH BAR
                                                                ;CABLE MODE TESTING LEAVES BIT 0
                                                                ;OF BAR SET.THEREFORE MUST
                                                                ;CHECK FOR ODD ADDRESS
011136  001401                          BEQ     .+4             ;IS THE BAR CORRECT?
011140  104000                          HLT                     ;BAR IS WRONG
011142  022777  052525  167722          CMP     #52525,@BDR     ;CHECK FOR CORRECT DATA
011150  001401                          BEQ     .+4             ;DATA GET TRANSFERRED?
011152  104000                          HLT                     ;BAD DATA IN BDR
011154  004737  010156                  JSR     PC,NORMAL
011160  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
011162  000404                          BR      TNPR0           ;GO TO NEXT TEST
011164  005077  167700          T33CLR: CLR     @CSR            ;CLEAR IE
011170  004737  010156                  JSR     PC,NORMAL
```

```
                                        .SBTTL TEST 56 TEST FOR 1 DATO NON BURST MODE TRANSFER
                                ;;*****************************************************************
                                ;       TEST 56 TEST FOR 1 DATO NON BURST MODE  TRANSFER
                                ;       CABLE MODE(WITH WRAP AROUND CABLE IN USER SLOTS)
                                ;;*****************************************************************
011174  000004                  TNPRO:  SCOPE
011176  005077  167666                  CLR     @CSR            ;FORCE TO BE CSR #1
011202  012777  177777  167654          MOV     #-1,@WCR        ;SET UP FOR 1 TRANSFER
011210  012777  001146  167650          MOV     #NPR1,@BAR      ;TRANSFER TO BUS ADDRESS IN NPR1
011216  005037  001146                  CLR     NPR1            ;GET READY TO RECEIVE DATA
011222  012777  052525  167642          MOV     #52525,@BDR     ;SET UP TO TRANSFER DATA
011230  012777  011304  167700          MOV     #INTC,@DRINV    ;INTERRUPT VECTOR TO INTC
011236  013777  001074  167674          MOV     DRINL,@DRVS     ;INTERRUPT STATUS TO LEVEL DRINL
011244  005077  167532                  CLR     @PSW            ;PROC STATUS TO ZERO
011250  012777  000112  167612          MOV     #112,@CSR       ;DATO(FNCT1),FNCT3,IE
011256  052777  000401  167604          BIS     #401,@CSR       ;CYCLE,GO
011264  005037  011272                  CLR     1$+2            ;WAIT FOR NPR AND INTER
011270  005227  000001          1$:     INC     #1
011274  001375                  BNE     1$
011276  104000                  HLT                             ;NO DR11-W INTERRUPT
011300  000005                  RESET                           ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                                ;INTERRUPT REQUEST IN BOARD HDWR.
011302  000424                  BR      T34CLR                  ;CLEAR IE
011304  004737  010246          INTC:   JSR     PC,ERRCHK       ;CLEAR IE,CHECK FOR ERROR
011310  005777  167550                  TST     @WCR            ;TEST WCR
011314  001401                  BEQ     .+4                     ;IS WCR EQUAL TO ZERO?
011316  104000                  HLT                             ;WCR EQUAL TO ZERO
011320  022777  001151  167540          CMP     #NPR1+3,@BAR    ;COMPARE CORRECT BAR WITH BAR
                                                                ;CABLE MODE TESTING LEAVES BIT 0
                                                                ;OF BAR SET.THEREFORE MUST
                                                                ;CHECK FOR ODD ADDRESS
011326  001401                  BEQ     .+4                     ;IS THE BAR CORRECT?
011330  104000                  HLT                             ;BAR IS WRONG
011332  023727  001146  052525          CMP     NPR1,#52525     ;CHECK FOR CORRECT DATA
011340  001401                  BEQ     .+4                     ;CORRECT DATA TRANSFERRED?
011342  104000                  HLT                             ;BAD DATA
011344  004737  010156                  JSR     PC,NORMAL
011350  022626                  CMP     (SP)+,(SP)+             ;RESTORE STACK
011352  000404                  BR      T37                     ;GO TO NEXT TEST
011354  005077  167510          T34CLR: CLR     @CSR            ;CLEAR IE
011360  004737  010156                  JSR     PC,NORMAL
```

```
                                        .SBTTL TEST 57 STRING OF 200 DATIS BURST MODE XFERS
                                ;;*****************************************************************
                                ;       TEST 57 STRING OF 200 DATI'S BURST MODE XFERS
                                ;       CABLE MODE(WITH WRAP AROUND CABLE IN USER SLOTS)
                                ;;*****************************************************************
011364  000004                T37:    SCOPE
011366  005077  167476              CLR     @CSR            ;FORCE TO BE CSR #1
011372  012737  000200  001156      MOV     #200,BUFLEN     ;LENGTH OF BUFFER=200
011400  004737  007714              JSR     PC,LODBUF       ;LOAD THE BUFFER WITH INCREMENTING PATTERN
011404  013737  001156  001164      MOV     BUFLEN,WCLEN    ;PREPARE NUMBER FOR WCR
011412  005437  001164              NEG     WCLEN           ;2'S COMPLEMENT OF BUFLEN
011416  013777  001164  167440      MOV     WCLEN,@WCR      ;SET UP WCR
011424  013777  001152  167434      MOV     INBUF,@BAR      ;SET UP BAR
011432  012777  177777  167432      MOV     #-1,@BDR        ;MAINT AIDE
011440  012777  011524  167470      MOV     #T37CHK,@DRINV  ;INT VECTOR
011446  013777  001074  167464      MOV     DRINL,@DRVS     ;INT VECTOR TO PRIORITY DRINL
011454  005077  167322              CLR     @PSW            ;LET THE DR11-W INTERRUPT
011460  012777  000100  167402      MOV     #100,@CSR       ;IE
011466  052777  000401  167374      BIS     #401,@CSR       ;CYCLE,GO
011474  012737  001000  001312      MOV     #1000,TIME      ;WAIT FOR INTERRUPT
011502  005337  001312          1$: DEC     TIME
011506  001375                      BNE     1$
011510  104000                      HLT
011512  000005                      RESET                   ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                            ;INTERRUPT REQUEST IN BOARD HDWR.
011514  004737  010156              JSR     PC,NORMAL
011520  000137  011544              JMP     T40
011524  022626              T37CHK: CMP     (SP)+,(SP)+
011526  004737  010016              JSR     PC,INTA         ;CLR IE,CHECK ERROR,READY,WC=0,&BAR
011532  022777  000177  167332      CMP     #177,@BDR       ;CHECK THAT WORD #200 OF INBUF IS IN BAR
011540  001401                      BEQ     .+4             ;IS IT?
011542  104000                      HLT                     ;BAD DATA IN BDR
```

```
2296                                            .SBTTL TEST 60 STRING OR 200 DATOS BURST MODE XFERS
2297                                    ;;******************************************************************
2298                                    ;       TEST 60 STRING OR 200 DATO'S BURST MODE XFERS
2299                                    ;       CABLE MODE(WITH WRAP AROUND CABLE IN USER SLOTS)
2300                                    ;;******************************************************************
2301 011544  000004                    T40:    SCOPE
2302 011546  005077  167316                    CLR     @CSR                    ;FORCE TO BE CSR #1
2303 011552  012737  000200  001156            MOV     #200,BUFLEN             ;LENGTH OF BUFFER=200
2304 011560  004737  007714                    JSR     PC,LODBUF               ;LOAD THE BUFFER WITH INCREMENTING PATTERN
2305 011564  013737  001156  001164            MOV     BUFLEN,WCLEN            ;PREPARE NUMBER FOR WCR
2306 011572  005437  001164                    NEG     WCLEN                   ;2'S COMPLEMENT OF BUFLEN
2307 011576  013777  001164  167260            MOV     WCLEN,@WCR              ,SET UP WCR
2308 011604  013777  001152  167254            MOV     INBUF,@BAR              ;SET UP BAR
2309 011612  012777  052525  167252            MOV     #52525,@BDR             ;SET UP BDR
2310 011620  012777  011704  167310            MOV     #T40CHK,@DRINV          ;INTERRUPT VECTOR
2311 011626  013777  001074  167304            MOV     DRINL,@DRVS             ;INTERRUPT VECTOR TO PRIORITY DRINL
2312 011634  005077  167142                    CLR     @PSW                    ;LET THE DR11-W INTERRUPT
2313 011640  012777  000102  167222            MOV     #102,@CSR               ;IE,FNCT1
2314 011646  052777  000401  167214            BIS     #401,@CSR               ;CYCLE,GO
2315 011654  012737  001000  001312            MOV     #1000,TIME              ;WAIT FOR INTERRUPT
2316 011662  005337  001312            1$:     DEC     TIME
2317 011666  001375                            BNE     1$
2318 011670  104000                            HLT
2319 011672  000005                            RESET                           ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
2320                                                                            ;INTERRUPT REQUEST IN BOARD HDWR.
2321 011674  004737  010156                    JSR     PC,NORMAL
2322 011700  000137  011716                    JMP     T56
2323 011704  022626                    T40CHK: CMP     (SP)+,(SP)+
2324 011706  004737  010016                    JSR     PC,INTA                 ;CLEAR IE,CHECK ERROR,READY,WC 0,& BAR
2325 011712  004737  010200                    JSR     PC,DATOCK               ;CHECK INBUF
```

```
                                        .SBTTL TEST 61 STRING OF 200 DATIS NON-BURST MODE
                                ;;*****************************************************************
                                ;       TEST 61 STRING OF 200 DATI'S NON-BURST MODE
                                ;       CABLE MODE(WITH WRAP AROUND CABLE IN USER SLOTS)
                                ;;*****************************************************************
011716  000004                T56:    SCOPE
011720  005077  167144              CLR     @CSR            ;FORCE TO BE CSR #1
011724  012737  000200  001156      MOV     #200,BUFLEN     ;LENGTH OF BUFFER=200
011732  004737  007714              JSR     PC,LODBUF       ;LOAD THE BUFFER WITH INCREMENTING PATTERN
011736  013737  001156  001164      MOV     BUFLEN,WCLEN    ;PREPARE NUMBER FOR WCR
011744  005437  001164              NEG     WCLEN           ;2'S COMPLEMENT OF BUFLEN
011750  013777  001164  167106      MOV     WCLEN,@WCR      ;SET-UP WCR
011756  013777  001152  167102      MOV     INBUF,@BAR      ;SET-UP BAR
011764  012777  177777  167100      MOV     #-1,@BDR        ;MAINT AIDE
011772  012777  012056  167136      MOV     #T56CHK,@DRINV  ;INT VECTOR
012000  013777  001074  167132      MOV     DRINL,@DRVS     ;INT VECTOR TO PRIORITY DRINL
012006  005077  166770              CLR     @PSW            ;LET THE DR11-W INTERRUPT
012012  012777  000110  167050      MOV     #110,@CSR       ;FNCT3,IE
012020  052777  000401  167042      BIS     #401,@CSR       ;CYCLE,GO
012026  012737  001000  001312      MOV     #1000,TIME      ;WAIT FOR INTERRUPT
012034  005337  001312      1$:     DEC     TIME
012040  001375                      BNE     1$
012042  104000                      HLT
012044  000005                      RESET                   ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                            ;INTERRUPT REQUEST IN BOARD HDWR.
012046  004737  010156              JSR     PC,NORMAL
012052  000137  012100              JMP     T57
012056  022626          T56CHK: CMP     (SP)+,(SP)+
012060  004737  010016              JSR     PC,INTA         ;CLEAR IE,CHECK ERROR,READY,WC 0,& BAR
012064  000240                      NOP
012066  022777  000177  166776      CMP     #177,@BDR       ;CHECK THAT WORD #200 OF INBUF IS IN BAR
012074  001401                      BEQ     .+4             ;IS IT?
012076  104000                      HLT                     ;BAD DATA IN BDR
```

```
2362                                            .SBTTL TEST 62 STRING OF 200 DATOS NON-BURST MODE
2363                                    ;;******************************************************************
2364                                    ;       TEST 62 STRING OF 200 DATO'S NON-BURST MODE
2365                                    ;       CABLE MODE(WITH WRAP AROUND CABLE IN USER SLOTS)
2366                                    ;******************************************************************
2367 012100  000004                    T57:    SCOPE
2368 012102  005077  166762                    CLR     @CSR            ;FORCE TO BE CSR #1
2369 012106  012737  000200  001156            MOV     #200,BUFLEN     ;LENGTH OF BUFFER-200
2370 012114  004737  007714                    JSR     PC,LODBUF       ;LOAD THE BUFFER WITH INCREMENTING PATTERN
2371 012120  013737  001156  001164            MOV     BUFLEN,WCLEN    ;PREPARE NUMBER FOR WCR
2372 012126  005437  001164                    NEG     WCLEN           ;2'S COMPLEMENT OF BUFLEN
2373 012132  013777  001164  166724            MOV     WCLEN,@WCR      ;SET UP WCR
2374 012140  013777  001152  166720            MOV     INBUF,@BAR      ;SET UP BAR
2375 012146  012777  052525  166716            MOV     #52525,@BDR     ;SET UP BDR
2376 012154  012777  012234  166754            MOV     #T57CHK,@DRINV  ;INTERRUPT VECTOR
2377 012162  013777  001074  166750            MOV     DRINL,@DRVS     ;INTERRUPT VECTOR TO PRIORITY DRINL
2378 012170  005077  166606                    CLR     @PSW            ;LET THE DR11-W INTERRUPT
2379 012174  012777  000102  166666            MOV     #102,@CSR       ;FNCT1,IE
2380 012202  052777  000401  166660            BIS     #401,@CSR       ;CYCLE,GO
2381 012210  012737  001000  001312            MOV     #1000,TIME      ;WAIT FOR INTERRUPT
2382 012216  005337  001312            1$:     DEC     TIME
2383 012222  001375                            BNE     1$
2384 012224  104000                            HLT
2385 012226  000005                            RESET                   ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
2386                                                                    ;INTERRUPT REQUEST IN BOARD HDWR.
2387 012230  004737  010156                    JSR     PC,NORMAL
2388 012234  022626                    T57CHK: CMP     (SP)+,(SP)+
2389 012236  004737  010016                    JSR     PC,INTA         ;CLEAR IE,CHECK ERROR,READY,WC 0,& BAR
2390 012242  000240                            NOP
2391 012244  004737  010200                    JSR     PC,DATOCK       ;CHECK INBUF
2392
```

```
                                        .SBTTL TEST 63 DETERMINE N-CYCLE SWITCH SETTING
                                ;;******************************************************************************
                                ;       TEST 63 DETERMINE N-CYCLE SWITCH SETTING
                                ;       HAVE SWITCH FLIPPED AND WAIT UNTIL IT IS DONE
                                ;       THIS IS DONE FIRST PASS ONLY
                                ;;******************************************************************************
012250  032737  000001  001144          BIT     #BIT0,BORW
012256  001550                          BEQ     SWT1
012260  005737  001200                  TST     FLAG            ;WERE WE HERE BEFORE
012264  001402                          BEQ     10$             ;NO
012266  000137  013132                  JMP     END             ;YES DONE THIS PASS
012272  005237  001200          10$:    INC     FLAG            ;SET FLAG FOR NEXT PASS
012276  012777  100000  166564          MOV     #BIT15,@CSR     ;GO TO EIR
012304  032777  000400  166556          BIT     #BIT8,@CSR      ;IS IT SET
012312  001055                          BNE     FLOP            ;YES
012314  012737  077777  001312  FLIP:   MOV     #77777,TIME     ;SET TIMER
012322  012705  000100                  MOV     #100,R5         ;SET PRINT DELAY
012326  104401  012334                  TYPE    ,65$            ;;TYPE ASCIZ STRING
012332  000425                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/CHANGE POSITION OF N-CYCLE BURST SWITCH/
012406                          64$:
012406  032777  000400  166454  20$:    BIT     #BIT8,@CSR      ;CHECK FOR CHANGE
012414  001402                          BEQ     30$             ;NOT YET
012416  000137  012600                  JMP     SWT1            ;DONE CONTINUE
012422  005337  001312          30$:    DEC     TIME            ;DEC TIMER
012426  001367                          BNE     20$             ;CHECK AGAIN
012430  012737  077777  001312          MOV     #77777,TIME     ;RESET TIMER
012436  005305                          DEC     R5              ;DEC PRINT DELAY
012440  001362                          BNE     20$             ;CHECK AGAIN
012442  000137  012314                  JMP     FLIP            ;TELL AGAIN TO FLIP SWITCH
012446  012737  077777  001312  FLOP:   MOV     #77777,TIME     ;SET TIMER
012454  012705  000100                  MOV     #100,R5         ;SET PRINT DELAY
012460  104401  012466                  TYPE    ,65$            ;;TYPE ASCIZ STRING
012464  000425                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/CHANGE POSITION OF N-CYCLE BURST SWITCH/
012540                          64$:
012540  032777  000400  166322  40$:    BIT     #BIT8,@CSR      ;CHECK FOR CHANGE
012546  001002                          BNE     50$             ;NOT YET
012550  000137  012600                  JMP     SWT1            ;DONE CONTINUE
012554  005337  001312          50$.    DEC     TIME            ;DEC TIMER
012560  001767                          BEQ     40$             ;CHECK AGAIN
012562  012737  077777  001312          MOV     #77777,TIME     ;RESET TIMER
012570  005305                          DEC     R5              ;DEC PRINT DELAY
012572  001362                          BNE     40$             ;CHECK AGAIN
012574  000137  012446                  JMP     FLOP            ;TELL AGAIN TO FLIP SWITCH
```

```
                                        .SBTTL TEST 64 STRING OF 200 DATIS BURST MODE XFERS (SWITCH FLIPPED)
                                ;;*****************************************************************************
                                ;       TEST 64 STRING OF 200 DATI'S BURST MODE XFERS (SWITCH FLIPPED)
                                ;       CABLE MODE(WITH WRAP AROUND CABLE IN USER SLOTS)
                                ;;*****************************************************************************
012600  000004                  SWT1:   SCOPE
012602  005077  166262                  CLR     @CSR            ;FORCE TO BE CSR #1
012606  012737  000200  001156          MOV     #200,BUFLEN     ;LENGTH OF BUFFER=200
012614  004737  007714                  JSR     PC,LODBUF       ;LOAD THE BUFFER WITH INCREMENTING PATTERN
012620  013737  001156  001164          MOV     BUFLEN,WCLEN    ;PREPARE NUMBER FOR WCR
012626  005437  001164                  NEG     WCLEN           ;2'S COMPLEMENT OF BUFLEN
012632  013777  001164  166224          MOV     WCLEN,@WCR      ;SET UP WCR
012640  013777  001152  166220          MOV     INBUF,@BAR      ;SET UP BAR
012646  012777  177777  166216          MOV     #-1,@BDR        ;MAINT AIDE
012654  012777  012740  166254          MOV     #SWINT1,@DRINV  ;INT VECTOR
012662  013777  001074  166250          MOV     DRINL,@DRVS     ;INT VECTOR TO PRIORITY DRINL
012670  005077  166106                  CLR     @PSW            ;LET THE DR11-W INTERRUPT
012674  012777  000100  166166          MOV     #100,@CSR       ;IE
012702  052777  000401  166160          BIS     #401,@CSR       ;CYCLE,GO
012710  012737  001000  001312          MOV     #1000,TIME      ;WAIT FOR INTERRUPT
012716  005337  001312          1$:     DEC     TIME
012722  001375                          BNE     1$
012724  104000                          HLT
012726  000005                          RESET                   ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                                ;INTERRUPT REQUEST IN BOARD HDWR.
012730  004737  010156                  JSR     PC,NORMAL
012734  000137  012760                  JMP     SWT2
012740  022626                  SWINT1: CMP     (SP)+,(SP)+
012742  004737  010016                  JSR     PC,INTA         ;CLR IE,CHECK ERROR,READY,WC=0,&BAR
012746  022777  000177  166116          CMP     #177,@BDR       ;CHECK THAT WORD #200 OF INBUF IS IN BAR
012754  001401                          BEQ     .+4             ;IS IT?
012756  104000                          HLT                     ;BAD DATA IN BDR
```

```
                                        .SBTTL TEST 65 STRING OR 200 DATOS BURST MODE XFERS (SWITCH FLIPPED)
                                ;;*****************************************************************************
                                ;       TEST 65 STRING OR 200 DATO'S BURST MODE XFERS (SWITCH FLIPPED)
                                ;       CABLE MODE(WITH WRAP AROUND CABLE IN USER SLOTS)
                                ;;*****************************************************************************
012760  000004                  SWT2:   SCOPE
012762  005077  166102                  CLR     @CSR            ;FORCE TO BE CSR #1
012766  012737  000200  001156          MOV     #200,BUFLEN     ;LENGTH OF BUFFER=200
012774  004737  007714                  JSR     PC,LODBUF       ;LOAD THE BUFFER WITH INCREMENTING PATTERN
013000  013737  001156  001164          MOV     BUFLEN,WCLEN    ;PREPARE NUMBER FOR WCR
013006  005437  001164                  NEG     WCLEN           ;2'S COMPLEMENT OF BUFLEN
013012  013777  001164  166044          MOV     WCLEN,@WCR      ;SET UP WCR
013020  013777  001152  166040          MOV     INBUF,@BAR      ;SET UP BAR
013026  012777  052525  166036          MOV     #52525,@BDR     ;SET UP BDR
013034  012777  013120  166074          MOV     #SWINT2,@DRINV  ;INTERRUPT VECTOR
013042  013777  001074  166070          MOV     DRINL,@DRVS     ;INTERRUPT VECTOR TO PRIORITY DRINL
013050  005077  165726                  CLR     @PSW            ;LET THE DR11-W INTERRUPT
013054  012777  000102  166006          MOV     #102,@CSR       ;IE,FNCT1
013062  052777  000401  166000          BIS     #401,@CSR       ;CYCLE,GO
013070  012737  001000  001312          MOV     #1000,TIME      ;WAIT FOR INTERRUPT
013076  005337  001312          1$:     DEC     TIME
013102  001375                          BNE     1$
013104  104000                          HLT
013106  000005                          RESET                   ;IF NO INTERRUPT,CLEAR ANY OUTSTANDING
                                                                ;INTERRUPT REQUEST IN BOARD HDWR.
013110  004737  010156                  JSR     PC,NORMAL
013114  000137  013132                  JMP     END
013120  022626                  SWINT2: CMP     (SP)+,(SP)+
013122  004737  010016                  JSR     PC,INTA         ;CLEAR IE,CHECK ERROR,READY,WC-0,& BAR
013126  004737  010200                  JSR     PC,DATOCK       ;CHECK INBUF

                                        ;THIS ENDS CABLE MODE TESTING
```

```
                                        ;;*****************************************************************************
                                        ;;      END OF PASS
                                        ;;*****************************************************************************
013132  012737  000207  177566  END:    MOV     #207,@#177566   ;RING BELL
013140  105737  177564                  TSTB    @#177564
013144  100375                          BPL     .-4
013146  005277  165642                  INC     @PNTPAS         ;INCREMENT PASS COUNT FOR THIS BOARD
013152  005737  001316                  TST     HLTDET          ;ERROR OCCUR THIS PASS?
013156  001402                          BEQ     2$              ;NO
013160  005277  165632                  INC     @PNTERR         ;YES INCR. ERROR COUNT FOR THIS BRD.
013164  032737  000001  001314  2$:     BIT     #1,CABLE        ;HAVE CABLE TESTS BEEN DONE?
013172  001405                          BEQ     1$              ;NO
013174  012702  014755                  MOV     #MANCBL,R2      ;YES
013200  004737  015372                  JSR     PC,TTOUT
013204  000404                          BR      END1
013206  012702  014723          1$:     MOV     #MANT,R2
013212  004737  015372                  JSR     PC,TTOUT
013216                          END1:
013216  012702  014622                  MOV     #$ENPAS,R2      ;PRINT 'END PASS'
013222  004737  015372                  JSR     PC,TTOUT
013226  017746  165562                  MOV     @PNTPAS,-(SP)   ;;SAVE @PNTPAS FOR TYPEOUT
013232  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
013234  104401  001354                  TYPE    ,$ENULL
013240  013702  000042                  MOV     @#42,R2
013244  001405                          BEQ     NXTBRD
013246  000005                          RESET
013250  004712                  $ENDAD: JSR     PC,(R2)
013252  000240                          NOP
013254  000240                          NOP
013256  000240                          NOP
013260  062737  000002  001010  NXTBRD: ADD     #2,PNTRAD       ;POINT TO NEXT BOARD FOR NEXT PASS
013266  062737  000002  001012          ADD     #2,PNTVAD
013274  062737  000002  001014          ADD     #2,PNTPAS
013302  062737  000002  001016          ADD     #2,PNTERR
013310  000137  001604                  JMP     BEGIN
```

```
                                        ;;*****************************************************************************
                                        ;       ENTERED WITH SYSTEM TRAP CALL(HLT)
                                        ;       PRINT OUT THE ERROR PC AND STATUS REGISTER
                                        ;;*****************************************************************************
013314  032777  000001  165546  PRINT:  BIT     #BIT0,@CSR      ;CHECK WHICH CSR
013322  001422                          BEQ     10$             ;WE ARE IN CSR #1
013324  017737  165540  001142          MOV     @CSR,EIR        ;SAVE CSR #2
013332  013701  001070                  MOV     CSR,R1          ;ADDRESS OF CSR INTO R1
013336  062701  000001                  ADD     #1,R1           ;R1 NOW HAS ADD OF UPPER BYTE
013342  142701  000200                  BICB    #200,R1         ;RETURN TO CSR #1
013346  017705  165516                  MOV     @CSR,R5         ;SAVE CSR IN R5
013352  105737  001143                  TSTB    EIR+1           ;TEST FOR ANY ERROR
013356  001406                          BEQ     20$             ;NO ERROR BIT(S) SET
013360  052705  100000                  BIS     #BIT15,R5       ;ERROR WAS SET - RESTORE IT
013364  000137  013374                  JMP     20$             ;CONTINUE
013370  017705  165474          10$:    MOV     @CSR,R5         ;SAVE CSR IN R5
013374  012737  000001  001316  20$:    MOV     #1,HLTDET       ;SAYS ERROR OCCURRED THIS PASS
013402  004737  014144                  JSR     PC,CKSWR        ;
013406  037727  165714  020000          BIT     @SWR,#20000     ;TEST FOR INHIBIT PRINT OUT
013414  001142                          BNE     1$              ;IF SO, BRANCH OVER PRINT
013416  012637  013770                  MOV     (SP)+,SAVPC     ;PC OF FAILING ROUTINE
013422  012637  013772                  MOV     (SP)+,SAVCC     ;CC OF ERROR CONDITION
013426  024646                          CMP     -(SP),-(SP)     ;REPOSITION THE STACK
013430  012777  000215  165540          MOV     #215,@TPB       ;CR
013436  105777  165532                  TSTB    @TPS
013442  100375                          BPL     .-4
013444  012777  000212  165524          MOV     #212,@TPB       ;LINE FEED
013452  105777  165516                  TSTB    @TPS
013456  100375                          BPL     .-4
013460  010237  013762                  MOV     R2,SAVR2        ;SAVE R2
013464  010337  013764                  MOV     R3,SAVR3        ;SAVE R3
013470  010437  013766                  MOV     R4,SAVR4        ;SAVE R4
013474  104401  001351                  TYPE    ,$CRLF
013500  032737  000001  001144          BIT     #BIT0,BORW
013506  001403                          BEQ     15$
013510  104401  015014                  TYPE    ,MSG4
013514  000402                          BR      25$
013516  104401  015163          15$:    TYPE    ,MSG7
013522  104401  001351          25$:    TYPE    ,$CRLF
013526  013746  001320                  MOV     TESTNO,-(SP)    ;;SAVE TESTNO FOR TYPEOUT
013532  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013534  104401  001354                  TYPE    ,$ENULL
013540  012777  000240  165430          MOV     #240,@TPB
013546  013746  013770                  MOV     SAVPC,-(SP)     ;;SAVE SAVPC FOR TYPEOUT
013552  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013554  104401  001354                  TYPE    ,$ENULL
013560  012777  000240  165410          MOV     #240,@TPB
013566  105777  165402                  TSTB    @TPS            ;SPACE BETWEEN WORDS
013572  100375                          BPL     .-4
013574  013746  013772                  MOV     SAVCC,-(SP)     ;;SAVE SAVCC FOR TYPEOUT
013600  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013602  104401  001354                  TYPE    ,$ENULL
013606  012777  000240  165362          MOV     #240,@TPB       ;PRINT SPACE
013614  105777  165354                  TSTB    @TPS            ;PRINTER DONE
013620  100375                          BPL     .-4             ;BRANCH WHEN NOT DONE
013622  012737  013650  000004          MOV     #3$,BUSERR      ;
013630  010546                          MOV     R5,-(SP)        ;;SAVE R5 FOR TYPEOUT
```

```
        013632  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2597   013634  104401  001354                  TYPE    ,$ENULL
 2598   013640  012737  000006  000004          MOV     #6,BUSERR
 2599   013646  000405                          BR      4$
 2600   013650  022626                  3$:     CMP     (SP)+,(SP)+
 2601   013652  012737  000006  000004          MOV     #6,BUSERR
 2602   013660  000401                          BR      5$
 2603   013662  000240                  4$:     NOP
 2604   013664  005737  001314          5$:     TST     CABLE           ;PRINT CABLE MSSGE?
 2605   013670  001404                          BEQ     2$              ;NO
 2606   013672  104401  015057                  TYPE    ,MSG5
 2607   013676  104401  015131                  TYPE    ,MSG6
 2608   013702  013702  013762          2$:     MOV     SAVR2,R2
 2609   013706  013703  013764                  MOV     SAVR3,R3
 2610   013712  013704  013766                  MOV     SAVR4,R4
 2611   013716  004737  014144                  JSR     PC,CKSWR
 2612   013722  005777  165400          1$:     TST     @SWR            ;CHECK SR FOR HALT SWITCH
 2613   013726  100001                          BPL     .+4
 2614   013730  000000                          HALT                    ;HALT ON ERROR UP IN SWR
 2615   013732  023737  000042  000046          CMP     @#42,@#46       ;ARE WE IN ACT11 AUTO MODE?
 2616   013740  001001                          BNE     .+4             ;BRANCH ON NO
 2617   013742  000000                          HALT                    ;HALT ON ERROR IF IN ACT11 AUTO MODE
 2618   013744  037727  165356  000400          BIT     @SWR,#400       ;GO TO NEXT BOARD?
 2619   013752  001402                          BEQ     CONTIN          ;IF NO,CONTINUE THIS BOARD
 2620   013754  000137  013132                  JMP     END
 2621
 2622   013760  000002                  CONTIN: RTI
 2623   013762  000000                  SAVR2:  0
 2624   013764  000000                  SAVR3:  0
 2625   013766  000000                  SAVR4:  0
 2626   013770  000000                  SAVPC:  0
 2627   013772  000000                  SAVCC:  0
 2628
```

```
                                ;;*****************************************************************
                                ;       SCOPE LOOP ROUTINE ENTERED BY USER TRAP
                                ;;*****************************************************************

013774  004737  014144          SCOPEA: JSR     PC,CKSWR
014000  032777  040000  165320          BIT     #40000,@SWR
014006  001003                        BNE     SCOPEB          ;SCOPE, BIT IS A ONE
014010  011637  014126                MOV     @SP,RETURN      ;NO - SAVE PC FOR NEXT TIME
014014  000002                        RTI                     ;RETURN IN SEQUENCE
014016  022606                  SCOPEB: CMP     (SP)+,SP        ;REPOSITION THE STACK
014020  012677  164756                MOV     (SP)+,@PSW
014024  000177  000076                JMP     @RETURN         ;SCOPE RETURN
```

```
                                ;;*****************************************************************************
                                ;       SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES
                                ;;*****************************************************************************

014030  004737  014144          SCOPEC: JSR     PC,CKSWR
014034  032777  040000  165264          BIT     #40000,@SWR     ;TEST SR FOR SCOPE
014042  001365                        BNE     SCOPEB          ;YES SCOPE
014044  005777  164744                TST     @PNTPAS         ;FIRST @PASS (@PASCNT-0) ?
014050  001415                        BEQ     SCOPEG          ;BR IF YES, INHIBIT ITERATIONS
014052  004737  014144                JSR     PC,CKSWR
014056  032777  004000  165242        BIT     #4000,@SWR      ;TEST FOR ITERATION
014064  001007                        BNE     SCOPEG          ;INHIBIT ITERATION
014066  023737  014124  014122        CMP     SCOPEF,ICOUNT
014074  001403                        BEQ     SCOPEG          ;EXIT - DONE
014076  005237  014124                INC     SCOPEF          ;INCREMENT COUNT
014102  000745                        BR      SCOPEB          ;LOOP SOME MORE
014104  005037  014124          SCOPEG: CLR     SCOPEF          ;CLEAR COUNT
014110  011637  014126                MOV     @SP,RETURN      ;SAVE SCOPE RETURN POINTER
014114  005237  001320                INC     TESTNO
014120  000002                        RTI                     ;RETURN INLINE-NEXT TEST
014122  004000          ICOUNT: 4000
014124  000000          SCOPEF: 0                               ;COUNT LOCATION FOR ITERATION LOOP
014126  001604          RETURN: BEGIN                           ;ADDRESS OF LAST TEST
                                .EVEN
014130  000137  000200                JMP     200
```

```
                                ;;*****************************************************************************
                                ;       CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
                                ;       OF LOC. 176.
                                ;;*****************************************************************************
014134  000000                  TEMPST: .WORD   0
014136  000000                  COUNT:  .WORD   0
014140  000000                  RDSW:   .WORD   0
014142  000000                  T  .    .WORD   0

014144  105777  165020          CKSWR:  TSTB    @TKS            ;YES,WAIT FOR
014150  100152                  BPL     OUT             ;READY, GET CHARACTER
014152  017737  165014  014142          MOV     @TKB,TIB        ;AND STRIP OFF
014160  042737  177600  014142          BIC     #177600,TIB     ;THE GARBAGE
014166  005737  000042                  TST     @#42            ;IF AUTO MODE,DON'T CHECK
014172  001012                  BNE     1$              ;FOR CNTRL S
014174  022737  000023  014142          CMP     #23,TIB
014202  001006                  BNE     1$              ;NOT CNTRL S
014204  012702  014560                  MOV     #$CNTS,R2
014210  004737  015372                  JSR     PC,TTOUT
014214  000137  021230                  JMP     RUNSUM          ;LEAVE PROGRAM AND PRINT RUN SUMMARY
014220  022737  000176  001326  1$:     CMP     #SWREG,SWR      ;SOFTW. SWITCH REG PRESENT?
014226  001123                  BNE     OUT

014230  022737  000007  014142          CMP     #7,TIB          ;IS IT A <^G>
014236  001117                  BNE     OUT
014240  012702  014552                  MOV     #$CNTG,R2
014244  004737  015372                  JSR     PC,TTOUT
014250  012702  014573          CNTLU:  MOV     #OUTSWR,R2
014254  004737  015372                  JSR     PC,TTOUT
014260  017746  165042                  MOV     @SWR,-(SP)      ;;SAVE @SWR FOR TYPEOUT
014264  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
014266  104401  001354                  TYPE    ,$ENULL
014272  012702  014603                  MOV     #INNEW,R2
014276  004737  015372                  JSR     PC,TTOUT
014302  005037  014134                  CLR     @#TEMPST
014306  005037  014134          $READ:  CLR     TEMPST
014312  012737  000007  014136          MOV     #7,COUNT
014320  004737  014500          1$:     JSR     PC,TTIN         ;GO  READ A CHARACTER
014324  042737  177600  014142          BIC     #177600,TIB     ;STRIP OFF GARBAGE
014332  122737  000025  014142          CMPB    #25,TIB         ;IS IT A ^U?
014340  001001                  BNE     2$              ;BRANCH IF NOT
014342  000742                  3$:     BR      CNTLU           ;START OVER
014344  122737  000015  014142  2$:     CMPB    #15,TIB         ;IS IT A <CR>?
014352  001011                  BNE     4$              ;BRANCH IF NOT
014354  012702  014567                  MOV     #CARLF,R2
014360  004737  015372                  JSR     PC,TTOUT
014364  022737  000007  014136          CMP     #7,COUNT        ;WAS IT FIRST CHARACTER
014372  001036                  BNE     7$              ;CHANGE SWR IF NOT FIRST ONE
014374  000440                  8$:     BR      OUT             ;GET OUT
014376  122737  000060  014142  4$:     CMPB    #60,TIB
014404  003004                  BGT     5$
014406  122737  000067  014142          CMPB    #67,TIB
014414  002005                  BGE     6$
014416  012702  014614          5$:     MOV     #$QUEST,R2
014422  004737  015372                  JSR     PC,TTOUT
014426  000745                  BR      3$              ;START OVER IF NOT LEGAL CHARACTER
```

```
 2729 014430  006337  014134          6$:     ASL     TEMPST
 2730 014434  006337  014134                  ASL     TEMPST
 2731 014440  006337  014134                  ASL     TEMPST
 2732 014444  142737  000060  014142          BICB    #60,TIB         ;GET NITTY-GRITTY
 2733 014452  153737  014142  014134          BISB    TIB,TEMPST
 2734 014460  005337  014136                  DEC     COUNT           ;ONLY WANT 6   DIGITS
 2735 014464  001754                          BEQ     5$
 2736 014466  000714                          BR      1$
 2737 014470  013777  014134  164630  7$:     MOV     TEMPST,@SWR     ;CHANGE SWITCH REGISTER CONTENTS
 2738 014476  000207                  OUT:    RTS     PC              ;RETURN TO PROGRAM
```

```
                                ;;*****************************************************************
                                ;       TTY READ SUBROUTINE*********
                                ;;*****************************************************************


014500  005077  164464          TTIN:   CLR     @TKS
014504  005077  164462                  CLR     @TKB
014510  005037  014142                  CLR     TIB
014514  005277  164450                  INC     @TKS
014520  105777  164444          TTIN1:  TSTB    @TKS
014524  100375                  TTIN1   BPL     TTIN1
014526  017737  164440  014142          MOV     @TKB,TIB
014534  105777  164434          TTIN2:  TSTB    @TPS
014540  100375                  TTIN2   BPL     TTIN2
014542  113777  014142  164426          MOVB    TIB,@TPB

014550  000207                          RTS     PC
014552     137     136     107  $CNTG.  .ASCIZ  '_^G &'
014555     040     046     000
014560     040     137     136  $CNTS:  .ASCIZ  / _^S &/
014563     123     040     046
014566     000
014567     137     040     046  CARLF:  .ASCIZ  '_ &'
014572     000
014573     137     123     127  OUTSWR: .ASCIZ  '_SWR= &'
014576     122     075     040
014601     046     000
014603     040     040     116  INNEW:  .ASCIZ  '  NEW- &'
014606     105     127     075
014611     040     046     000
014614     137     077     040  $QUEST: .ASCIZ  '_? _&'
014617     137     046     000
014622     040     137     040  $ENPAS: .ASCIZ  ' _ END PASS   #   &'
014625     105     116     104
014630     040     120     101
014633     123     123     040
014636     040     040     043
014641     040     040     040
014644     046     000
014646     137     040     103  $TITLE: .ASCIZ  '_ CZDRL-DR11W GEN NPR INTFC LOGIC TEST   __&'
014651     132     104     122
014654     114     055     104
014657     122     061     061
014662     127     040     107
014665     105     116     040
014670     116     120     122
014673     040     111     116
014676     124     106     103
014701     040     114     117
014704     107     111     103
014707     040     124     105
014712     123     124     040
014715     040     040     137
014720     137     046     000
014723     137     124     105  MAINT:  .ASCIZ  /_TEST MODE: MAINT ONLY  &/
014726     123     124     040
```

```
        014731  115  117  104
        014734  105  072  040
        014737  115  101  11'
        014742  116  124  040
        014745  117  116  114
        014750  131  040  040
        014753  046  000
  2767  014755  015  012  124  MANCBL: .ASCIZ<15><12>/TEST MODE: MAINT AND CABLE/<15><12>
        014760  105  123  124
        014763  040  115  117
        014766  104  105  072
        014771  040  115  101
        014774  111  116  124
        014777  040  101  116
        015002  104  040  103
        015005  101  102  114
        015010  105  015  012
        015013  000
  2768  015014  124  105  123  MSG4:   .ASCIZ  /TESTNO  ERRPC  PSW   DR11W STATUS /
        015017  124  116  117
        015022  040  040  105
        015025  122  122  120
        015030  103  040  040
        015033  120  123  127
        015036  040  040  040
        015041  104  122  061
        015044  061  127  040
        015047  123  124  101
        015052  124  125  123
        015055  040  000
  2769  015057  015  012  120  MSG5: .ASCIZ <15><12>/PROGRAM IS PERFORMING CABLE MODE TEST/<15><12>
        015062  122  117  107
        015065  122  101  115
        015070  040  111  123
        015073  040  120  105
        015076  122  106  117
        015101  122  115  111
        015104  116  107  040
        015107  103  101  102
        015112  114  105  040
        015115  115  117  104
        015120  105  040  124
        015123  105  123  124
        015126  015  012  000
  2770  015131  125  123  105  MSG6: .ASCIZ /USER CABLE SHOULD BE IN/<15><12>
        015134  122  040  103
        015137  101  102  114
        015142  105  040  123
        015145  110  117  125
        015150  114  104  040
        015153  102  105  040
        015156  111  116  015
        015161  012  000
  2771  015163  015  012  124  MSG7: .ASCIZ <15><12>/TESTNO  ERRPC  PSW  DR11B STATUS/<15><12>
        015166  105  123  124
        015171  116  117  040
        015174  040  105  122
```

```
                015177  122     120     103
                015202  040     040     120
                015205  123     127     040
                015210  040     104     122
                015213  061     061     102
                015216  040     123     124
                015221  101     124     125
                015224  123     015     012
                015227  000
   2772         015230  015     012     104     MSG8: .ASCIZ <15><12>/DIAGNOSTIC TESTING OF DR11B NOW IN PROGRESS/<15><12>
                015233  111     101     107
                015236  116     117     123
                015241  124     111     103
                015244  040     124     105
                015247  123     124     111
                015252  116     107     040
                015255  117     106     040
                015260  104     122     061
                015263  061     102     040
                015266  116     117     127
                015271  040     111     116
                015274  040     120     122
                015277  117     107     122
                015302  105     123     123
                015305  015     012     000
   2773         015310  015     012     104     MSG9: .ASCIZ <15><12>/DIAGNOSTIC TESTING OF DR11W NOW IN PROGRESS/<15><12>
                015313  111     101     107
                015316  116     117     123
                015321  124     111     103
                015324  040     124     105
                015327  123     124     111
                015332  116     107     040
                015335  117     106     040
                015340  104     122     061
                015343  061     127     040
                015346  116     117     127
                015351  040     111     116
                015354  040     120     122
                015357  117     107     122
                015362  105     123     123
                015365  015     012     000
   2774                                         .EVEN
   2775
   2776
   2777         015370  000000          OFL:    0                       ;FIRST CHAR FLAG
   2778
   2779
```

```
                                        ;;*****************************************************************************
                                        ;       TTY ASCII OUTPUT ROUTINE
                                        ;;*****************************************************************************


015372  105712                  TTOUT:  TSTB    (R2)            ;CHECK FOR NULL CHARACTER
015374  001403                          BEQ     1$              ;IF NOT, TYPE THE CHARACTER
015376  122712  000046                  CMPB    #'&,(R2)        ;CHECK FOR TERMINATOR
015402  001005                          BNE     .EMPTY
015404  042777  000100  163562  1$:     BIC     #100,@TPS
015412  005002                          CLR     R2              ;CLEAR POINTER TO CHARACTER
015414  000207                          RTS     PC              ;RETURN
015416  122712  000137          .EMPTY: CMPB    #'_,(R2)        ;CRLF CHAR?
015422  001411                          BEQ     .RET
015424  122712  000041                  CMPB    #'!,(R2)        ;CHECK FOR RETURN TERMINATOR
015430  001414                          BEQ     .REST
015432  105777  163536          1$:     TSTB    @TPS
015436  100375                          BPL     1$
015440  112277  163532                  MOVB    (R2)+,@TPB      ;TYPE CHARACTER
015444  000752                          BR      TTOUT
015446  005202                  .RET:   INC     R2
015450  010237  021402                  MOV     R2,.SAV         ;SET UP NEW POINTER
015454  012702  015470                  MOV     #.RETR,R2
015460  000767                          BR      .RET-6
015462  013702  021402          .REST:  MOV     .SAV,R2
015466  000741                          BR      TTOUT

015470     015     012     041  .RETR:  .BYTE   15,12,'
                                        .EVEN
```

```
                                        .SBTTL
                                        .SBTTL  SYSMAC ROUTINES
                                .SBTTL  TYPE ROUTINE
                                ;;******************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*
015474  105737  001343          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
015500  100002                          BPL     1$              ;;BR IF YES
015502  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
015504  000407                          BR      3$              ;;LEAVE
015506  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
015510  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
015514  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
015516  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
015520  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
015522  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
015524  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
015530  000002                          RTI                     ;;RETURN
015532  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
015536  001430                          BEQ     8$
015540  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
015544  001006                          BNE     5$
015546  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
015550  104401                          TYPE                    ;;TYPE A CR AND LF
015552  001351                          $CRLF
015554  105037  015710                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
015560  000755                          BR      2$              ;;GET NEXT CHARACTER
015562  004737  015644          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
015566  123726  001342          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
015572  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
015574  013746  001340                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
015600  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
015604  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
015606  004737  015644                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
015612  105337  015710                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
015616  000770                          BR      7$              ;;LOOP
                                ;HORIZONTAL TAB PROCESSOR
015620  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
015624  004737  015644          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
015630  132737  000007  015710          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
015636  001372                          BNE     9$              ;;TAB STOP
015640  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
015642  000724                          BR      2$              ;;GET NEXT CHARACTER
015644  105777  163464          $TYPEC: TSTB    @$$TPS          ;;WAIT UNTIL PRINTER IS READY
015650  100375                          BPL     $TYPEC
```

```
015652  116677  000002  163456          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
015660  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
015666  001003                          BNE     1$              ;;BRANCH IF NO
015670  105037  015710                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
015674  000406                          BR      $TYPEX          ;;EXIT
015676  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
015704  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
015706  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
015710  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
015712  000207          $TYPEX: RTS     PC
```

```
2815                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
                                        ;;**************************************************************
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                        ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M               ;;M-1 OR 0
                                        ;*                                      ;;1-TYPE LEADING ZEROS
                                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                        ;*
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
015714  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
015720  116637  000001  016137          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
015726  112637  016141                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
015732  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
015736  000406                          BR      $TYPON
015740  112737  000001  016137  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
015746  112737  000006  016141          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
015754  112737  000005  016136  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
015762  010346                          MOV     R3,-(SP)        ;;SAVE R3
015764  010446                          MOV     R4,-(SP)        ;;SAVE R4
015766  010546                          MOV     R5,-(SP)        ;;SAVE R5
015770  113704  016141                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
015774  005404                          NEG     R4
015776  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
016002  110437  016140                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
016006  113704  016137                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
016012  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
016016  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
016020  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
016022  000404                          BR      3$              ;;GO DO MSB
016024  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
016026  006105                          ROL     R5
016030  006105                          ROL     R5
016032  010503                          MOV     R5,R3
016034  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
016036  105337  016140                  DECB    $OMODE          ;;TYPE THIS DIGIT?
016042  100016                          BPL     7$              ;;BR IF NO
016044  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
016050  001002                          BNE     4$              ;;TEST FOR 0
016052  005704                          TST     R4              ;;SUPPRESS THIS 0?
016054  001403                          BEQ     5$              ;;BR IF YES
016056  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
016060  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
016064  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
```

```
016070  110337  016134                  MOVB    R3,8$           ;;SAVE FOR TYPING
016074  104401  016134                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
016100  105337  016136          7$:     DECB    $OCNT           ;;COUNT BY 1
016104  003347                          BGT     2$              ;;BR IF MORE TO DO
016106  002402                          BLT     6$              ;;BR IF DONE
016110  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
016112  000744                          BR      2$              ;;GO DO THE LAST DIGIT
016114  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
016116  012604                          MOV     (SP)+,R4        ;;RESTORE R4
016120  012603                          MOV     (SP)+,R3        ;;RESTORE R3
016122  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
016130  012616                          MOV     (SP)+,(SP)
016132  000002                          RTI                     ;;RETURN
016134     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
016135     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
016136     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
016137     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
016140  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
2817                                    .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
                                        ;;**********************************************************************
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                        ;*REPLACED WITH SPACES.
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE
016142                                  $TYPDS:
016142  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
016144  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
016146  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
016150  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
016152  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
016154  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
016160  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
016164  100004                          BPL     1$              ;;BR IF INPUT IS POS.
016166  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
016170  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
016176  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
016200  012703  016356                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
016204  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
016210  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
016212  016001  016346                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
016216  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
016220  002402                          BLT     4$              ;;BR IF DONE
016222  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
016224  000774                          BR      3$
016226  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
016230  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
016232  001002                          BNE     5$              ;;FALL THROUGH IF 0
016234  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
016236  100407                          BMI     7$              ;;BR IF YES
016240  106316                  5$:     ASLB    (SP)            ;;MSD?
016242  103003                          BCC     6$              ;;BR IF NO
016244  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
016252  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
016256  052702  000C40          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
016262  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
016264  005720                          TST     (R0)+           ;;JUST INCREMENTING
016266  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
016272  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
016274  003002                          BGT     8$              ;;GO TO EXIT
016276  010502                          MOV     R5,R2           ;;GET THE LSD
016300  000764                          BR      6$              ;;GO CHANGE TO ASCII
016302  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
016304  100003                          BPL     9$              ;;BR IF NO
016306  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
016314  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
016316  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
016320  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
016322  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
016324  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
016326  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
016330  104401  016356                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
```

```
016334  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
016342  012616                          MOV     (SP)+,(SP)
016344  000002                          RTI                     ;;RETURN TO USER
016346  023420                  $DTBL:  10000.
016350  001750                          1000.
016352  000144                          100.
016354  000012                          10.
016356                          $DBLK:  .BLKW   4
```

```
2819                                    .SBTTL  TTY INPUT ROUTINE
                                        ;;*****************************************************************
                                        .ENABL  LSB
                                        .DSABL  LSB
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                        ;*CALL:
                                        ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                        ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                        ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                        ;
016366  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
016370  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
016376  105777  162726          1$:     TSTB    @$TKS           ;;WAIT FOR
016402  100375                          BPL     1$              ;;A CHARACTER
016404  117766  162722  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
016412  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
016420  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
016426  001013                          BNE     3$              ;;BRANCH IF NO
016430  105777  162674          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
016434  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
016436  117746  162670                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
016442  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
016446  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
016452  001366                          BNE     2$              ;;IF NOT DISCARD IT
016454  000750                          BR      1$              ;;YES, RESUME
016456  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
016464  002407                          BLT     4$              ;;BRANCH IF YES
016466  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
016474  003003                          BGT     4$              ;;BRANCH IF YES
016476  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
016504  000002                  4$:     RTI                     ;;GO BACK TO USER
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                        ;*CALL:
                                        ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                        ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                        ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
016506  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
016510  012703  016614          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
016514  022703  016623          2$:     CMP     #$TTYIN+7.,R3   ;;BUFFER FULL?
016520  101405                          BLOS    4$              ;;BR IF YES
016522  104406                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
016524  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
016526  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
016532  001003                          BNE     3$              ;;SKIP IF NOT
016534  104401  001350          4$:     TYPE    ,$QUES          ;;TYPE A '?'
016540  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
016542  111337  016612          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
016546  104401  016612                  TYPE    ,9$
016552  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
016556  001356                          BNE     2$              ;;LOOP IF NOT RETURN
016560  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
016564  104401  001352                  TYPE    ,$LF            ;;TYPE A LINE FEED
016570  012603                          MOV     (SP)+,R3        ;;RESTORE R3
016572  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
016574  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
```

```
016602  012766  016614  000004          MOV     #$TTYIN,4(SP)
016610  000002                          RTI                             ;;RETURN
016612     000                  9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
016613     000                          .BYTE   0                       ;;TERMINATOR
016614                          $TTYIN: .BLKB   7.                      ;;RESERVE 7. BYTES FOR TTY INPUT
016623     136     125     015  $CNTLU: .ASCIZ  /^U/<15><12>            ;;CONTROL 'U'
016626     012     000
016630     136     107     015  $CNTLG: .ASCIZ  /^G/<15><12>            ;;CONTROL 'G'
016633     012     000
016635     015     012     123  $MSWR:  .ASCIZ  <15><12>/SWR = /
016640     127     122     040
016643     075     040     000
016646     040     040     116  $MNEW:  .ASCIZ  /  NEW - /
016651     105     127     040
016654     075     040     000
                        .EVEN
```

```
2821                                    .SBTTL  READ A DECIMAL NUMBER FROM THE TTY
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL READ A DECIMAL (ASCII) NUMBER FROM THE TTY AND
                                        ;*CHANGE IT TO BINARY. IF TOO MANY CHARACTERS OR ANY ILLEGAL CHARACTERS
                                        ;*ARE READ A "?" FOLLOWED BY A CARRIAGE RETURN-LINE FEED WILL BE TYPED.
                                        ;*THE COMPLETE NUMBER MUST BE RETYPED. THE INPUT IS TERMINATED BY THE
                                        ;*USER TYPING A CARRIAGE RETURN. THE RANGE OF THE INPUT NUMBER IS
                                        ;*POSITIVE 32767 TO NEGATIVE 32768.
                                        ;*CALL:
                                        ;*      RDDEC                   ;;READ A DECIMAL NUMBER
                                        ;*      RETURN HERE             ;;NUMBER IS ON TOF OF THE STACK
                                        ;
016660  011646                  $RDDEC: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR
016662  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE INPUT NUMBER
016670  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
016672  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
016674  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
016676  104407                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
016700  012600                          MOV     (SP)+,R0        ;;ADDRESS OF 1ST CHAR.
016702  010037  017026                  MOV     R0,6$           ;;SAVE INCASE OF BAD INPUT
016706  005046                          CLR     -(SP)           ;;CLEAR DATA WORD
016710  005002                          CLR     R2              ;;SIGN SET POSITIVE
016712  122710  000055                  CMPB    #'-,(R0)        ;;SEE IF A MINUS SIGN WAS TYPED
016716  001001                          BNE     2$              ;;BR IF NO MINUS SIGN
016720  112002                          MOVB    (R0)+,R2        ;;SAVE FOR LATER USE
016722  112001                  2$:     MOVB    (R0)+,R1        ;;PICKUP THIS CHARACTER
016724  001424                          BEQ     3$              ;;GET OUT IF ZERO
016726  122701  000060                  CMPB    #'0,R1          ;;MAKE SURE THIS CHARACTER
016732  003032                          BGT     5$              ;;IS A DIGIT BETWEEN 0 & 9
016734  122701  000071                  CMPB    #'9,R1
016740  002427                          BLT     5$
016742  032716  170000                  BIT     #^C7777,(SP)    ;;DON'T LET NUMBER GET TO BIG
016746  001024                          BNE     5$              ;;BR IF NUMBER WOULD OVERFLOW
016750  006316                          ASL     (SP)            ;;*2
016752  011646                          MOV     (SP),-(SP)      ;;SAVE FOR LATER
016754  006316                          ASL     (SP)            ;;*4
016756  006316                          ASL     (SP)            ;;*8.
016760  062616                          ADD     (SP)+,(SP)      ;;*10.
016762  102416                          BVS     5$              ;;OVERFLOW ISN'T ALLOWED
016764  162701  000060                  SUB     #'0,R1          ;;STRIP AWAY THE ASCII JUNK
016770  060116                          ADD     R1,(SP)         ;;ADD IN THIS DIGIT
016772  102412                          BVS     5$              ;;OVERFLOW ISN'T ALLOWED
016774  000752                          BR      2$              ;;LOOP
016776  005702                  3$:     TST     R2              ;;CHECK IF NUMBER IS NEG
017000  001401                          BEQ     4$              ;;BR IF NO
017002  005416                          NEG     (SP)            ;;YES--NEGATE THE NUMBER
017004  012666  000012          4$:     MOV     (SP)+,12(SP)    ;;SAVE THE RESULT
017010  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
017012  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
017014  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
017016  000002                          RTI                     ;;RETURN
017020  005726                  5$:     TST     (SP)+           ;;CLEAN PARTIAL NUMBER FROM STACK
017022  105010                          CLRB    (R0)            ;;SET A TERMINATOR
017024  104401                          TYPE                    ;;TYPE THE INPUT UP TO BAD CHAR.
017026  000000                  6$:     .WORD   0               ;;POINTER GOES HERE
017030  104401  001350                  TYPE    ,$QUES          ;;"?" "CR" &"LF"
017034  000720                          BR      1$              ;;TRY AGAIN
```

```
2823                                    .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                        ;*CHANGE IT TO BINARY.
                                        ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                                        ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
                                        ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                                        ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
                                        ;*CALL:
                                        ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                                        ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                        ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
017036  011646                  $RDOCT: MOV     (SP),-(SP)              ;;PROVIDE SPACE FOR THE
017040  016666  000004  000002          MOV     4(SP),2(SP)             ;;INPUT NUMBER
017046  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
017050  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
017052  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
017054  104407                  1$:     RDLIN                           ;;READ AN ASCIZ LINE
017056  012600                          MOV     (SP)+,R0                ;;GET ADDRESS OF 1ST CHARACTER
017060  010037  017164                  MOV     R0,5$                   ;;AND SAVE IT
017064  005001                          CLR     R1                      ;;CLEAR DATA WORD
017066  005002                          CLR     R2
017070  112046                  2$:     MOVB    (R0)+,-(SP)             ;;PICKUP THIS CHARACTER
017072  001420                          BEQ     3$                      ;;IF ZERO GET OUT
017074  122716  000060                  CMPB    #'0,(SP)                ;;MAKE SURE THIS CHARACTER
017100  003026                          BGT     4$                      ;;IS AN OCTAL DIGIT
017102  122716  000067                  CMPB    #'7,(SP)
017106  002423                          BLT     4$
017110  006301                          ASL     R1                      ;;*2
017112  006102                          ROL     R2
017114  006301                          ASL     R1                      ;;*4
017116  006102                          ROL     R2
017120  006301                          ASL     R1                      ;;*8
017122  006102                          ROL     R2
017124  042716  177770                  BIC     #^C7,(SP)               ;;STRIP THE ASCII JUNK
017130  062601                          ADD     (SP)+,R1                ;;ADD IN THIS DIGIT
017132  000756                          BR      2$                      ;;LOOP
017134  005726                  3$:     TST     (SP)+                   ;;CLEAN TERMINATOR FROM STACK
017136  010166  000012                  MOV     R1,12(SP)               ;;SAVE THE RESULT
017142  010237  017174                  MOV     R2,$HIOCT
017146  012602                          MOV     (SP)+,R2                ;;POP STACK INTO R2
017150  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
017152  012600                          MOV     (SP)+,R0                ;;POP STACK INTO R0
017154  000002                          RTI                             ;;RETURN
017156  005726                  4$:     TST     (SP)+                   ;;CLEAN PARTIAL FROM STACK
017160  105010                          CLRB    (R0)                    ;;SET A TERMINATOR
017162  104401                          TYPE                            ;;TYPE UP THRU THE BAD CHAR.
017164  000000                  5$:     .WORD   0
017166  104401  001350                  TYPE    ,$QUES                  ;;"?" "CR" & "LF"
017172  000730                          BR      1$                      ;;TRY AGAIN
017174  000000                  $HIOCT: .WORD   0                       ;;HIGH ORDER BITS GO HERE
```

```
 2825                                   .SBTTL  TRAP DECODER
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE ''TRAP'' INSTRUCTION
                                        ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                        ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                        ;*GO TO THAT ROUTINE.
 017176  010046                         $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 017200  016600  000002                         MOV     2(SP),R0        ;;GET TRAP ADDRESS
 017204  005740                                 TST     -(R0)           ;;BACKUP BY 2
 017206  111000                                 MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 017210  006300                                 ASL     R0              ;;POSITION FOR INDEXING
 017212  016000  017232                         MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 017216  000200                                 RTS     R0              ;;GO TO ROUTINE
                                        ;;THIS IS USE TO HANDLE THE ''GETPRI'' MACRO
 017220  011646                         $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 017222  016666  000004  000002                 MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 017230  000002                                 RTI                     ;;RESTORE THE PSW
                                        .SBTTL  TRAP TABLE
                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE ''TRAP'' INSTRUCTION.
                                        ;       ROUTINE
                                        ;       -------
 017232  017220                         $TRPAD: .WORD   $TRAP2
 017234  015474                                 $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 017236  015740                                 $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 017240  015714                                 $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 017242  015754                                 $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 017244  016142                                 $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 017246  016366                                 $RDCHR  ;;CALL=RDCHR    TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
 017250  016506                                 $RDLIN  ;;CALL=RDLIN    TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
 017252  017036                                 $RDOCT  ;;CALL=RDOCT    TRAP+10(104410) READ AN OCTAL NUMBER FROM TTY
 017254  016660                                 $RDDEC  ;;CALL=RDDEC    TRAP+11(104411) READ A DECIMAL NUMBER FROM TTY
 2876
```

```
                                        .SBTTL  MULTIPLE BOARD DIALOGUE


                                ;       MBD MONITOR-COMMAND DECODER

                                MBD:
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF><CRLF>/DO YOU WISH MULTIPLE BOARD DIALOGUE?/
                                64$:
                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                        BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <CRLF>/(Y=YES, N=NO):    /
                                66$:
                                        RDCHR
                                        MOV     (SP)+,ANSWER
                                        CMP     ANSWER,#131     ;IS ANSWER YES?
                                        BEQ     YESMBD  ;IF YES,CONTINUE
                                        CMP     ANSWER,#116     ;IS ANSWER NO?
                                        BNE     ERRMSG          ;IF NEITHER YES OR NO ERROR IN REPLY
                                        TYPE    ,69$            ;;TYPE ASCIZ STRING
                                        BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  /N/<CRLF>
                                68$:

                                        RTS     PC              ; NO,SKIP DIALOGUE
                                ERRMSG:
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  /  ????  /
                                64$:
                                        BR MBD

                                YESMBD:
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  /Y/<CRLF><CRLF>
                                64$:
                                PROMPT:
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/=  /
                                64$:
                                        RDCHR                   ;
                                        MOV     (SP)+,ANSWER
                                        CMP     ANSWER,#114     ;LIST PRESENT TABLE?
                                        BEQ     LSTTBL  ;YES
                                        CMP     ANSWER,#122     ;RUN PROGRAM?
                                        BEQ     RUNPRG          ;YES
                                        CMP     ANSWER,#105     ;EDIT TABLE?
                                        BEQ     EDIT

                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                        BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  ???  /
                                66$:
                                        BR PROMPT
```

```
                                                LSTTBL:
                                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                        BR      64$             ;;GET OVER THE ASCIZ
                                                ;;65$:  .ASCIZ  /L/
                                                64$:
                                                        JSR     PC,PRTBL
                                                        BR      PROMPT
                                                RUNPRG:
                                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                        BR      64$             ;;GET OVER THE ASCIZ
                                                ;;65$:  .ASCIZ  /R/<CRLF>
                                                64$:
                                                        RTS     PC
                                                EDIT:
                                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                        BR      64$             ;;GET OVER THE ASCIZ
                                                ;;65$:  .ASCIZ  /E/
                                                64$:
                                                        JSR     PC,TBLEDT
                                                        BR      PROMPT

                                                PRTBL:  CLR     LOOP
                                                        MOV     #REGADR,R1
                                                        MOV     #VECADR,R2
                                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                        BR      64$             ;;GET OVER THE ASCIZ
                                                ;;65$:  .ASCIZ  <CRLF><CRLF><CRLF>/NO. BOARDS REQUESTED FOR TESTING:  /
                                                64$:
                                                        MOV     QTYBRD,-(SP)    ;;SAVE QTYBRD FOR TYPEOUT
                                                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                                        BR      66$             ;;GET OVER THE ASCIZ
                                                ;;67$:  .ASCIZ  <CRLF><CRLF>/BOARD#  REGSTR.  VECTOR/<CRLF>
                                                66$:
                                                        INC     LOOP
                                                1$:
                                                        TYPE    ,69$            ;;TYPE ASCIZ STRING
                                                        BR      68$             ;;GET OVER THE ASCIZ
                                                ;;69$:  .ASCIZ  <CRLF>
                                                68$:
                                                        MOV     LOOP,-(SP)      ;;SAVE LOOP FOR TYPEOUT
                                                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                                                        TYPE    ,71$            ;;TYPE ASCIZ STRING
                                                        BR      70$             ;;GET OVER THE ASCIZ
                                                ;;71$:  .ASCIZ  /  /
                                                70$:
                                                        MOV     (R1)+,-(SP)     ;;SAVE (R1)+ FOR TYPEOUT
                                                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                                        TYPE    ,73$            ;;TYPE ASCIZ STRING
                                                        BR      72$             ;;GET OVER THE ASCIZ
                                                ;;73$:  .ASCIZ  /  /
                                                72$:
                                                        MOV     (R2)+,-(SP)     ;;SAVE (R2)+ FOR TYPEOUT
                                                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                                        CMP     LOOP,QTYBRD
                                                        BEQ     PRTFIN
```

```
 2885 020056  005237  001322                 INC     LOOP
 2886 020062  000744.                        BR      1$.
 2887 020064                          PRTFIN:
      020064  104401  020072                 TYPE    ,65$            ;;TYPE ASCIZ STRING
      020070  000402                         BR      64$             ;;GET OVER THE ASCIZ
                                     ;;65$:  .ASCIZ  <CRLF><CRLF>
      020076                         64$:
 2888 020076  000207                         RTS     PC
 2889
 2890 020100  012701  001024        TBLEDT: MOV     #REGADR,R1
 2891 020104  012702  001076                MOV     #VECADR,R2
 2892 020110  005037  001322                CLR LOOP
 2893 020114                          BRDNO:
      020114  104401  020122                 TYPE    ,65$            ;;TYPE ASCIZ STRING
      020120  000425                         BR      64$             ;;GET OVER THE ASCIZ
                                     ;;65$:  .ASCIZ  <CRLF><CRLF>/HOW MANY BOARDS DO YOU WANT TO TEST?  /
      020174                         64$:
 2894 020174  104411                         RDDEC
 2895 020176  012637  001360                MOV     (SP)+,ANSWER
 2896 020202  005737  001360                TST     ANSWER
 2897 020206  001742                         BEQ     BRDNO
 2898 020210  023727  001360  000020         CMP     ANSWER,#20
 2899 020216  003423                         BLE     CORRCT
 2900 020220  104401  020226                 TYPE    ,67$            ;;TYPE ASCIZ STRING
      020224  000416                         BR      66$             ;;GET OVER THE ASCIZ
                                     ;;67$:  .ASCIZ  <CRLF>/MAX BOARDS OF 16 EXCEEDED/
      020262                         66$:
 2901 020262  000137  020536                JMP     EDTFIN
 2902 020266  013737  001360  001022  CORRCT: MOV     ANSWER,QTYBRD
 2903 020274  005237  001322                INC LOOP
 2904 020300                          ENTRY:
      020300  104401  020306                 TYPE    ,65$            ;;TYPE ASCIZ STRING
      020304  000405                         BR      64$             ;;GET OVER THE ASCIZ
                                     ;;65$:  .ASCIZ  <CRLF><CRLF>/BOARD /
      020320                         64$:
 2905 020320  013746  001322                MOV     LOOP,-(SP)      ;;SAVE LOOP FOR TYPEOUT
      020324  104405                         TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 2906 020326  104401  020334                 TYPE    ,67$            ;;TYPE ASCIZ STRING
      020332  000413                         BR      66$             ;;GET OVER THE ASCIZ
                                     ;;67$:  .ASCIZ  <CRLF>/ REGISTER ADDRESS: /
      020362                         66$:
 2907 020362  011146                         MOV     (R1),-(SP)      ;;SAVE (R1) FOR TYPEOUT
      020364  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2908 020366  104401  020374                 TYPE    ,69$            ;;TYPE ASCIZ STRING
      020372  000402                         BR      68$             ;;GET OVER THE ASCIZ
                                     ;;69$:  .ASCIZ  /   /
      020400                         68$:
 2909 020400  104410                         RDOCT
 2910 020402  012637  001360                MOV     (SP)+,ANSWER
 2911 020406  005737  001360                TST     ANSWER
 2912 020412  001403                         BEQ     2$
 2913
 2914 020414  013721  001360                MOV     ANSWER,(R1)+    ;INSTALL NEW # IN TABLE
 2915 020420  000401                         BR      3$
 2916 020422  005721                 2$:     TST     (R1)+
 2917 020424                          3$:
      020424  104401  020432                 TYPE    ,71$            ;;TYPE ASCIZ STRING
```

```
        020430  000411                          BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  /VECTOR ADDRFSS: /
        020454                          70$:
 2918   020454  011246                          MOV     (R2),-(SP)      ;;SAVE (R2) FOR TYPEOUT
        020456  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2919   020460  104401  020466                  TYPE    ,73$            ;;TYPE ASCIZ STRING
        020464  000402                          BR      72$             ;;GET OVER THE ASCIZ
                                        ;;73$:  .ASCIZ  /   /
        020472                          72$:
 2920   020472  104410                          RDOCT
 2921   020474  012637  001360                  MOV     (SP)+,ANSWER
 2922   020500  005737  001360                  TST     ANSWER
 2923   020504  001403                          BEQ     4$
 2924
 2925   020506  013722  001360                  MOV     ANSWER,(R2)+
 2926   020512  000401                          BR      5$
 2927   020514  005722                  4$:     TST     (R2)+
 2928
 2929   020516  023737  001322  001022  5$:     CMP     LOOP,QTYBRD
 2930   020524  001404                          BEQ     EDTFIN
 2931   020526  005237  001322                  INC     LOOP
 2932   020532  000137  020300                  JMP     ENTRY
 2933   020536  000207                  EDTFIN: RTS     PC
```

```
                                        .SBTTL  BURST DATA LATE CALIBRATION ROUTINE

                                        BRSTDL:
020540  104401  020546                          TYPE    ,65$            ;;TYPE ASCIZ STRING
020544  000426                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <CRLF><CRLF>/DO YOU WISH BURST DATA LATE CALIBRATION?/
020622                                  64$:
020622  104401  020630                          TYPE    ,67$            ;;TYPE ASCIZ STRING
020626  000412                                  BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <CRLF>/(Y=YES, N=NO):   /
020654                                  66$:
020654  104406                                  RDCHR
020656  012637  001360                          MOV     (SP)+,ANSWER
020662  023727  001360  000131                  CMP     ANSWER,#131     ;YES?
020670  001421                                  BEQ     3$
020672  023727  001360  000116                  CMP     ANSWER,#116     ;NO?
020700  001006                                  BNE     4$
020702  104401  020710                          TYPE    ,69$            ;;TYPE ASCIZ STRING
020706  000402                                  BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  /N/<CRLF>
020714                                  68$:
020714  000207                                  RTS     PC
020716                                  4$:
020716  104401  020724                          TYPE    ,71$            ;;TYPE ASCIZ STRING
020722  000403                                  BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  / ??? /
020732                                  70$:
020732  000702                                  BR      BRSTDL

                                        3$:
020734  104401  020742                          TYPE    ,73$            ;;TYPE ASCIZ STRING
020740  000402                                  BR      72$             ;;GET OVER THE ASCIZ
                                        ;;73$:  .ASCIZ  /Y/<CRLF>
020746                                  72$:
020746  104401  020754                          TYPE    ,75$            ;;TYPE ASCIZ STRING
020752  000426                                  BR      74$             ;;GET OVER THE ASCIZ
                                        ;;75$:  .ASCIZ  <CRLF>/BURST DATA LATE CALIBRATION IN PROGRESS../
021030                                  74$:
021030  104401  021036                          TYPE    ,77$            ;;TYPE ASCIZ STRING
021034  000414                                  BR      76$             ;;GET OVER THE ASCIZ
                                        ;;77$:  .ASCIZ  <CRLF>/ATTACH SCOPE PROBE.../
021066                                  76$:
021066  104401  021074                          TYPE    ,79$            ;;TYPE ASCIZ STRING
021072  000424                                  BR      78$             ;;GET OVER THE ASCIZ
                                        ;;79$:  .ASCIZ  <CRLF>/TO ABORT THIS ROUTINE,HALT PROCESSOR../
021144                                  78$:
021144  013700  001004                          MOV     DEFRAD,R0       ;BURST DATA LATE CALIBR. USES
                                                                        ;DEFAULT REG. AND VECTOR ADDRESSES
021150  022020                                  CMP     (R0)+,(R0)+
021152  010037  001070                          MOV     R0,CSR
021156  013700  001006                          MOV     DEFVAD,R0
021162  010037  001136                          MOV     R0,DRINV
021166  005720                                  TST     (R0)+
021170  010037  001140                          MOV     R0,DRVS
021174  004737  010156                          JSR     PC,NORMAL
021200  005077  157664                  2$:     CLR     @CSR            ;CLR CYCLE BIT
021204  012737  000100  001312                  MOV     #100,TIME
```

```
                                        BIS     #400,@CSR       ;SET CYCLE BIT
                                1$:     DEC     TIME            ;WAIT
                                        BNE     1$
                                        BR      2$              ;LOOP ALWAYS:SETTING AND CLEARING CYCLE
                                                                ;TO ABORT,HALT PROCESSOR


                                .SBTTL  RUN SUMMARY ROUTINE

                                RUNSUM: MOV     QTYBRD,R0               ;NO. OF BOARDS TESTED
                                        MOV     #REGADR,R1
                                        MOV     #PASCNT,R2
                                        MOV     #ERRCNT,R3
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/RUN SUMMARY.../<CRLF>
                                64$:
                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                        BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <CRLF>/BOARD  PASCNT  ERRCNT/<CRLF>
                                66$:
                                1$:
                                        MOV     (R1)+,-(SP)     ;;SAVE (R1)+ FOR TYPEOUT
                                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                        MOV     (R2)+,-(SP)     ;;SAVE (R2)+ FOR TYPEOUT
                                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                                        MOV     (R3)+,-(SP)     ;;SAVE (R3)+ FOR TYPEOUT
                                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                                        TYPE    ,$ENULL
                                        DEC     R0
                                        BEQ     2$
                                        TYPE    ,69$            ;;TYPE ASCIZ STRING
                                        BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <CRLF>
                                68$:
                                        BR      1$
                                2$:     RESET
                                        JMP     200
                                .SAV:   0
                                        .=.+1000
                                BUFF:   0                       ;FOR STACK POINTER 100 LOCATIONS
                                XINBUF: .
                                        .-.+1000
                                XCHKBU: .
                                LEVEL:  .WORD 0
                                        .END
```

```
ANSWER  001360
ATTN  = 020000
BAOFCK  010546
BAR     001066
BDFAIL  001362
BDR     001072
BEGAUT  001670
BEGIN   001604
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BORW    001144
BRDNO   020114
BRDSET  001656
BRLEV   006272
BRSTDL  020540
BRWAIT  001162
BUFF    022404
BUFLEN  001156
BUSERR  000004
CABLE   001314
CARLF   014567
CBL     010622
CHKA    007764
CHKBFF  007752
CHKBUF  001154
CKBAR   002116
CKBDR   002142
CKCSR   002130
CKDRIN  002154
CKDRVS  002170
CKFIN   002336
CKSWR   014144
CKWCR   002104
CNTLU   014250
COMPAR  010132
COMPRR  010214
CONTIN  013760
CORRCT  020266
COUNT   014136
CPUERR  002204
CR    = 000015
CRLF  = 000200
CSR     001070
CYCLE = 000400
DATCHK  010116
DATOCK  010200
DDISP = 177570
DEFRAD  001004
DEFVAD  001006
DIOMEM  001150
DISPLA  001324
DISPRE  000174
DONE    005556
DRINL   001074
DRINV   001136
DRVS    001140
DSTATA= 001000
DSTATB= 002000
DSTATC= 004000
DSWR  = 177570
EDIT    017610
EDTFIN  020536
EIR     001142
END     013132
END1    013216
ENTRY   020300
ERRCHK  010246
ERRCNT  001246
ERRDO   006534
ERRDO1  006614
ERRMSG  017430
ERROR = 100000
ERRVEC= 000004
FLAG    001200
FLIP    012314
FLOP    012446
FNCCNT  001202
FNCT1 = 000002
FNCT2 = 000004
FNCT3 = 000010
GO    = 000001
HLT   = 104000
HLTDET  001316
HT    = 000011
ICOUNT  014122
IE    = 000100
INBUF   001152
INBUF1  001204
INCBA   004652
INCDB   005406
INCWC   004622
INNEW   014603
INTA    010016
INTB    011114
INTC    011304
LDEXIT  007750
LENCHK  001160
LEVEL   023412
LF    = 000012
LOAD    001712
LOADA   007726
LODBUF  007714
LOOP    001322
LSTTBL  017556
MAINT = 010000
MANCBL  014755
MANT    014723
MBD     017256
MFLOOP  006662
MSG4    015014
MSG5    015057
MSG6    015131
MSG7    015163
MSG8    015230
MSG9    015310
NEX   = 040000
NEXCHK  010406
NOCBL   010734
NOP   = 000240
NORMAL  010156
NPRRDY  011012
NPR1    001146
NRM     007514
NSTAT   006132
NXTBRD  013260
OFL     015370
OK      007370
OUT     014476
OUTSWR  014573
PASCNT  001206
PINV    006166
PNTERR  001016
PNTPAS  001014
PNTRAD  001010
PNTVAD  001012
PRGCYC  001020
PRINT   013314
PROMPT  017464
PRTBL   017626
PRTFIN  020064
PSW     001002
P3INT   005720
P3INV   005624
P7ERR   006100
P7INV   006010
QTYBRD  001022
RDCHR = 104406
RDDEC = 104411
RDLIN = 104407
RDOCT = 104410
RDSW    014140
RDYCHK  001166
READY = 000200
REGADR  001024
RETURN  014126
RSET    007504
RUNPRG  017574
RUNSUM  021230
R6    =%000006
R7    =%000007
SAVCC   013772
SAVPC   013770
SAVR2   013762
SAVR3   013764
SAVR4   013766
SCOP    010450
SCOPE = 000004
SCOPEA  013774
SCOPEB  014016
SCOPEC  014030
SCOPEF  014124
SCOPEG  014104
SR      001000
STACK = 022404
SUSWR   001364
SWINT1  012740
SWINT2  013120
SWR     001326
SWREG   000176
SWT1    012600
SWT2    012760
TBLEDT  020100
TEMP    001306
TEMPST  011134
TEMP2   000310
TESTNO  001320
TIB     014442
TIME    001312
TKB     001142
TKS     001170
TKVEC = 000060
TNPRO   011174
TNPR1   010770
TPB     001176
TPS     001174
TRAPVE= 000034
TSTRDY  005534
TTIN    014500
TTIN1   014520
TTIN2   014534
TTOUT   015372
TYPDS = 104405
TYPE  = 104401
TYPOC = 104402
TYPON = 104404
TYPOS = 104403
T1STR   002070
T26     004250
T27     004372
T3      002414
T33     010744
T33CLR  011164
T33STR  010764
T34CLR  011354
T37     011364
T37CHK  011524
T40     011544
T40CHK  011704
T41     005360
T42     010276
T55     006440
T56     011716
T56CHK  012056
T57     012100
T57CHK  012234
T60     006632
T60CHK  006762
T61     007024
T61CHK  007174
T61.5   007206
T62CHK  007676
VECADR  001076
WCLEN   001164
WCR     001064
WRONG1  007360
WRONG2  007510
XBA16 = 000020
XBA17 = 000040
XCHKBU  023410
XINBUF  022406
X1      006122
X4      005560
X5      004676
X7      005740
YESCBL  010722
YESMBD  017452
$BELL   001344
$CHARC  015710
$CNTG   014552
$CNTLG  016630
$CNTLU  016623
$CNTS   014560
$CRLF   001351
$DBLK   016356
$DTBL   016346
$ENDAD  013250
$ENPAS  014622
$ENULL  001354
$ILLC   001342
$FALLS  001341
$HD   = 000003
$HIOUT  017174
$LF     001352
$MNEW   016646
$MSWR   016635
$NULL   001340
$OCNT   016136
$OMODE  016140
$QUES   001350
$QUEST  014614
$RDCHR  016366
```

```
$RDDEC  016660          $SWR  = 160000          $TPS    001334          $TYPE   015474          $OFILL  016137
$RDLIN  016506          $TITLE  014646          $TRAP   017176          $TYPEC  015644          .EMPTY  015416
$RDOCT  017036          $TKB    001332          $TRAP2  017220          $TYPEX  015712          .REST   015462
$RDSZ = 000007          $TKS    001330          $TRP  = 000012          $TYPOC  015740          .REF    015446
$READ   014306          $TN   = 000001          $TRPAD  017232          $TYPON  015754          .RETR   015470
$SETUP= 000004          $TPB    001336          $TTYIN  016614          $TYPOS  015714          .SAV    021402
$STUP = 177777          $TPFLG  001343          $TYPDS  016142

. ABS.  023414      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  20701 WORDS  ( 81 PAGES)
DYNAMIC MEMORY:  20746 WORDS  ( 79 PAGES)
ELAPSED TIME:  00:02:15
DR11W,DR11W SYSMAC.MLB/ML,DR11W.P11
```