.REM_

IDENTIFICATION

PRODUCT CODE: AC-F419A-MC

PRODUCT NAME: CXMNBA0 MNCDI MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1.0 ABSTRACT

THE MNB IS AN IOMOD THAT EXERCISES THE MNCDI DIGITAL INPUT.
THE SOFTWARE MODULE CONSISTS OF THREE SECTIONS. THE FIRST IS
THE DEFAULT CONDITION WHEN SR1 = 0. READ-WRITE TESTS ARE PERFORMED
TO VERIFY THE INTERNAL DATA PATH'S OF THE MNCDI LOGIC.
THE SECOND IS EXECUTED WHEN A MNCDO OUTPUT IS CONNECTED TO THE
MNCDI INPUT. THIS PROVIDES ADDITIONAL TEST OF THE CONTROL
SIGNALS. THE THIRD IS A WRAP-AROUND DATA TEST VERIFING THE DATA
OUTPUT AND INPUT CIRCUITS. UP TO 8 MNCDI'S CAN BE EXERCISED
WITH THIS MODULE. IF SR1 = 1, BE SURE TO "DESELECT" THE MNE
(DIGITAL OUT) MODULE.

2.0 REQUIREMENTS

HARDWARE: ONE MNCDI (DIGITAL IN).
ONE MNCDO (DIGITAL OUT) <OPTIONAL>
STORAGE: MNB REQUIRES:
DECIMAL WORDS: 1573
OCTAL WORDS: 3045
OCTAL BYTES: 6112

3.0 PASS DEFINITION

WHEN SR1 = 0, ONE PASS OF THE MNB MODULE CONSISTS OF GENERATION
4000 (8) INTERRUPTS.
WHEN SR1 = 1, ONE PASS OF THE MNB MODULE CONSISTS OF GENERATING
5000 (8) INTERRUPTS.

4.0 EXECUTION TIME

WHEN SR1 = 0, ONE PASS OF THE MNB MODULE RUNNING ALONE TAKES
APPROXIMATELY ONE MINUTE.
WHEN SR1 = 1, ONE PASS OF THE MNB MODULE RUNNING ALONE TAKES
APPROXIMATELY TWO MINUTES.

5.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 171160, VECTOR 130, BR1: 4 DEVCNT: 1, SR1: 0

REQUIRED PARAMETERS:

NONE IF SR1 = 0 IF SR1 = 1 THEN:
$BASE1 MUST CONTAIN THE MNCDO BUS ADDRESS
$VECT1 MUST CONTAIN THE MNCDO INTERRUPT VECTOR AND BE
SURE TO DESELECT THE "MNE" (DIGITAL OUT) MODULE.

6.0 DEVICE/OUTPUT SET-UP

THE FRONT PANEL SWITCHES MUST BE IN THE "-" POSITION.
NO ADDITIONAL IF SR1 = 0 IF SR1 = 1 THEN:
THE WRAP-AROUND CABLE MUST BE INSTALLED TO A MNCDO AND
BE SURE TO DESELECT THE "MNE" (DIGITAL OUT) MODULE.

7.0 MODULE OPERATION

THE FOLLOWING TESTS ARE PERFORMED ON THE MNCDI (SR1=0)

FLOAT A 1 ACROSS THE STIMULUS BIT REGISTER
FLOAT A 0 ACROSS THE STIMULUS BIT REGISTER
BYTE OPERATION OF THE STIMULUS BIT REGISTER
READ-WRITE TESTS OF BITS 1 - 6,8,9,12 AND 14 IN THE STATUS REGISTER
BYTE OPERATION OF THE STATUS REGISTER
MAINT. STROBE SETS INPUT READY FLAG
INPUT READY CAN BE WRITTEN TO A ZERO
INPUT READY WILL NOT SET IF NO "SBR MATCH"
OVERRUN FLAG SETS
OVERRUN FLAG CAN BE WRITTEN TO A ZERO
INVERT DATA FUNCTIONS CORRECTLY
EACH BIT OF THE INPUT REGISTER CAN BE CLEARED
INPUT READY FLAG INTERRUPT TEST
OVERRUN FLAG INTERRUPT TEST

8.0 OPERATION OPTIONS

SR1 = 0 RUN MNCDI LOGIC TEST

SR1 = 1 RUN MNCDI LOGIC TEST
RUN MNCDO LOGIC TEST
RUN MNCDO TO MNCDI WRAPAROUND CONTROL TEST
RUN MNCDO TO MNCDI WRAPAROUND DATA TEST

IF DEVCNT (DVID1) CONTAINS MORE THAT A 1, MULTIPLE MNCDI
WILL BE TESTED. IF DEVCNT (DVID1) CONTAINS MORE THAN 1 AND SR1=1,
THERE MUST AT LEAST THE SAME NUMBER OF MNCDO'S CONNECTED TO MNCDI'S.
THE FIRST ADDRESS MNCDO UNIT MUST BE CONNECTED TO THE FIRST
ADDRESS MNCDI WITH ADDITIONAL UNITS ALSO PAIRED TOGETHER.

9.0 NON-STANDARD PRINTOUTS

ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11
DOCUMENT.

```
                                        .LIST   MF
                                        .NLIST  MC,CND,MD
                                        .TITLE  MNBA DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6      23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************
000000'                         BEGIN:
000000' 047115  040502      040 MODNAM: .ASCII  /MNBA / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   0       ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 171160                  ADDR:   171160+0        ;1ST DEVICE ADDR.
000010' 000130                  VECTOR: 130+0           ;1ST DEVICE VECTOR.
000012'     200                 BR1:    .BYTE   PRTY4+0 ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY0+0 ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1             ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN            ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN            ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN            ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN            ;SWITCH REGISTER 4
                                ;*****************************************************************
000026' 140000                  STAT:   140000          ;STATUS WORD.
000030' 000300'                 INIT:   START           ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP           ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0               ;PASS COUNTER.
000036' 002000                  ICONT:  2000            ;# OF ITERATIONS PER PASS=2000
000040' 000000                  ICOUNT: 0               ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0               ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0               ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0               ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0               ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0               ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0               ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                 ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0               ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0               ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN            ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN            ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN            ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN            ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN            ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN            ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN            ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN            ;ADDR OF CURRENT CSR.
000102'                         SBADR:                  ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN            ;CONTENTS OF CSR.
000104'                         WASADR:                 ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN            ;STATUS REG CONTENTS.
000106'                         ERRTYP:                 ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN            ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN            ;ACTUAL DATA.
000112' 000306'                 RSTRT:  RESTPT          ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN            ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN            ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN            ;# OF INTERRUPTS PER ITERATION
000122' 000000                  IDNUM:  0               ;MODULE IDENTIFICATION NUMBER=0
```



```
000224'                         MODSP:
                                ;*****************************************************************

000224' 171260                  $BASE:: 171260          ;INITIAL BUS ADDRESS OF THE MNCDO <IF SR1=1>
000226' 000340                  $VECT:: 340             ;INITIAL INTERRUPT VECTOR OF THE MNCDO <IF SR1=1>

                                ;OUTPUT ADDRESS
000230' 000000                  OCSR:   0
000232' 000000                  OCSR1:  0                       ;HIGH BYTE ADDRESS

000234' 000000                  DOR:    0
000236' 000000                  DOR1:   0                       ;HIGH BYTE ADDRESS

                                ;INPUT ADDRESS
000240' 000000                  ICSR:   0
000242' 000000                  ICSR1:  0                       ;HIGH BYTE ADDRESS
000244' 000000                  DIR:    0
000246' 000000                  DIR1:   0
000250' 000000                  SBR:    0
000252' 000000                  SBR1:   0                       ;HIGH BYTE ADDRESS

000254' 000000                  DODINV: 0
000256' 000000                  DODINS: 0

000260' 000000                  DIDINV: 0
000262' 000000                  DIDINS: 0
000264' 000000                  DIEINV: 0
000266' 000000                  DIEINS: 0
000270' 000000                  SRMINE: 0                       ;MY COPY OF SR1
000272' 000000                  TEMP:   0
000274' 000000                  TEMP1:  0
000276' 000001                  TEMP2:  BIT0
          010000                        BITDAT=BIT12            ;MAINT INPUT INHIBIT
          004000                        BITEXT=BIT11            ;MAINT INPUT STROBE
```

```
000300' 016767 177512 177762 START:  MOV     SR1,SRMINE      ;COPY SR1 FOR MY USE
                             ;INITILIZE THE BUS ADDRESSES AND VECTORS
000306' 012767 000001 177762 RESTRT: MOV     #BIT0,TEMP2     ;LOAD UNIT SELECT FLAG
000314' 012700 000230'       1$:     MOV     #DCSR,R0        ;LOAD ADDRESS POINTER
000320' 016701 177700                MOV     $BASE1,R1       ;LOAD INITIAL BUS ADDRESS
000324' 010120               1$:     MOV     R1,(R0)+        ;LOAD DEVICE ADDRESS
000326' 005201                       INC     R1              ;UPDATE BUS ADDRESS VALUE
000330' 020027 000240'               CMP     R0,#ICSR        ;TEST IF DONE WITH BUS ADDRESSES
000334' 001373                       BNE     1$              ;BR IF NOT
000336' 016701 177444                MOV     ADDR,R1         ;LOAD INPUT ADDRESS
000342' 010120               2$:     MOV     R1,(R0)+        ;LOAD THE ADDRESS
000344' 005201                       INC     R1              ;UPDATE ADDRESS
000346' 020027 000254'               CMP     R0,#DODINV      ;TEST IF AT END
000352' 001373                       BNE     2$              ;BR IF NOT
000354' 016701 177646                MOV     $VECT1,R1       ;LOAD VECTOR POINTER
000360' 010120               3$:     MOV     R1,(R0)+        ;LOAD DEVICE VECTOR ADDRESS
000362' 005721                       TST     (R1)+           ;UPDATE BUS VECTOR VALUE
000364' 020027 000260'               CMP     R0,#DODINS+2    ;TEST IF DONE WITH BUS VECTORS
000370' 001373                       BNE     3$              ;BR IF NOT
000372' 016701 177412                MOV     VECTOR,R1       ;LOAD INPUT VECTOR
000376' 010120               4$:     MOV     R1,(R0)+        ;LOAD THE VECTOR
000400' 005721                       TST     (R1)+           ;BUMP THE ADDRESS
000402' 020027 000270'               CMP     R0,#DIEINS+2    ;TEST IF DONE
000406' 001373                       BNE     4$              ;BR IF NOT

                             ;TEST IF UNIT IS TO RUN
000410' 036767 177662 177376 CONT1:  BIT     TEMP2,DVID1     ;IS UNIT SELECTED ?
000416' 001040                       BNE     INPUT           ;BR IF SELECTED
                             ;UNIT IS NOT SELECTED -- CORRECT THE ADDRESSES
000420' 012700 000230'       CONT2:  MOV     #DCSR,R0        ;GET ADDRESS
000424' 012701 000004                MOV     #4,R1           ;GET NEW OFFSET VALUE
000430' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000432' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000434' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000436' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000440' 012701 000010                MOV     #10,R1          ;RELOAD NEW OFFSET
000444' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000446' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000450' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000452' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000454' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000456' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000460' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000462' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000464' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000466' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000470' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
000472' 060110                       ADD     R1,(R0)         ;UPDATE THE VALUE
                             ;DETERMINE IF NEXT UNIT IS TO BE TESTED
000474' 006367 177576                ASL     TEMP2           ;CHANGE UNITS
000500' 022767 000400 177570         CMP     #BIT8,TEMP2     ;TEST IF LAST UNIT
000506' 001340                       BNE     CONT1           ;BR IF NOT
000510' 104413 000000'               ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                             ;MONITOR SHALL TEST END OF PASS
000514' 000167 177566                JMP     RESTRT
```



```
                             ;VERIFY A MNCDI BUS ADDRESS RESPONSE
000520' 016767 177514 177352 INPUT:  MOV     ICSR,CSRA               ;LOAD ADDR.
000526' 005777 177506                TST     @ICSR                   ;TEST INPUT STATUS
000532' 005777 177506                TST     @DIR                    ;TEST INPUT DATA REGISTER
000536' 005777 177506                TST     @SBR                    ;TEST STIM. BUFFER REGISTER
000542'                      DI1:
                             ;FLOAT A 1 ACROSS THE MNCDI STIMULUS BIT REGISTER
000542' 012767 000001 177332         MOV     #BIT0,ACSR              ;LOAD EXPECT BIT
000550' 016777 177326 177472 1$:     MOV     ACSR,@SBR               ;LOAD MNCDI STIMULUS BIT REGISTER
000556' 017767 177466 177320        MOV     @SBR,ASTAT              ;READ MNCDI STIMULUS BIT REGISTER
000564' 026767 177312 177312        CMP     ACSR,ASTAT              ;COMPARE
000572' 001403                       BEQ     2$                      ;BR IF SAME
                             ;*************************************************************
000574' 104405 000000' 000000        HRDER$,BEGIN,NULL       ; MNCDI STIMULUS BIT REGISTER FAILED TO HOLD A FLOATING
                             ;*************************************************************
000602'                      2$:
000602' 104407 000000'               BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
000606' 104407 000000'               BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000612' 006367 177264                ASL     ACSR                    ;CHANGE THE DATA
000616' 001354                       BNE     1$                      ;BR IF MORE DATA
000620'                      DI2:
                             ;FLOAT A 0 ACROSS THE MNCDI STIMULUS BIT REGISTER
000620' 012767 000001 177444         MOV     #BIT0,TEMP              ;LOAD INITIAL BIT
000626' 016767 177440 177246 1$:     MOV     TEMP,ACSR               ;LOAD EXPECTED
000634' 005167 177242                COM     ACSR                    ;COMPLEMENT
000640' 016777 177236 177402        MOV     ACSR,@SBR               ;LOAD MNCDI STIMULUS BIT REGISTER
000646' 017767 177376 177230        MOV     @SBR,ASTAT              ;READ MNCDI STIMULUS BIT REGISTER
000654' 026767 177222 177222        CMP     ACSR,ASTAT              ;COMPARE
000662' 001403                       BEQ     2$                      ;BR IF SAME
                             ;*************************************************************
000664' 104405 000000' 000000        HRDER$,BEGIN,NULL       ;MNCDI STIMULUS BIT REGISTER FAILED TO HOLD A FLOATING 0
                             ;*************************************************************
000672'                      2$:
000672' 104407 000000'               BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
000676' 104407 000000'               BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000702' 006367 177364                ASL     TEMP                    ;CHANGE THE DATA
000706' 001347                       BNE     1$                      ;BR IF MORE DATA
000710'                      DI3:
                             ;VERIFY BYTE OPERATION ON THE MNCDI STIMULUS BIT REGISTER
000710' 012777 177777 177332 1$:     MOV     #-1,@SBR                ;LOAD MNCDI STIMULUS BIT REGISTER
000716' 012767 000377 177156        MOV     #377,ACSR               ;LOAD EXPECTED
000724' 105077 177322                CLRB    @SBR1                   ;CLEAR HIGH BYTE
000730' 017767 177314 177146        MOV     @SBR,ASTAT              ;READ MNCDI STIMULUS BIT REGISTER
000736' 026767 177140 177140        CMP     ACSR,ASTAT              ;COMPARE
000744' 001403                       BEQ     2$                      ;BR IF SAME
                             ;*************************************************************
000746' 104405 000000' 000000        HRDER$,BEGIN,NULL       ;CLEARING HIGH BYTE CHANGED LOW BYTE
                             ;*************************************************************
000754' 012777 177777 177266 2$:     MOV     #-1,@SBR                ;LOAD MNCDI STIMULUS BIT REGISTER
000762' 012767 177400 177112        MOV     #177400,ACSR            ;LOAD EXPECTED
000770' 105077 177254                CLRB    @SBR                    ;CLEAR LOW BYTE
000774' 017767 177250 177102        MOV     @SBR,ASTAT              ;READ MNCDI STIMULUS BIT REGISTER
001002' 026767 177074 177074        CMP     ACSR,ASTAT              ;COMPARE
001010' 001403                       BEQ     3$                      ;BR IF SAME
                             ;*************************************************************
001012' 104405 000000' 000000        HRDER$,BEGIN,NULL       ;CLEARING LOW BYTE CHANGED HIGH BYTE
```

```
                                        ;*************************************************************
001020'                         3$:
001020' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
001024' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001030'                         DI4:
                                        ;TEST THAT BIT1 OF MNCDI STATUS REGISTER IS READ-WRITE
001030' 012767  000002  177044          MOV     #BIT1,ACSR              ;LOAD EXPECTED
001036' 016777  177040  177174          MOV     ACSR,@ICSR              ;LOAD BIT1 INTO MNCDI STATUS REGISTER
001044' 017767  177170  177032          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
001052' 042767  000200  177024          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
001060' 026767  177016  177016          CMP     ACSR,ASTAT              ;TEST THAT IT SET
001066' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
001070' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT1 OF MNCDI STATUS REGISTER FAILED TO SET
                                        ;*************************************************************
001076' 046777  177000  177134  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
001104' 017767  177130  176772          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
001112' 026767  176764  176764          CMP     ACSR,ASTAT              ;TEST THE BIT
001120' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*************************************************************
001122' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT1 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                        ;*************************************************************
001130'                         2$:
001130' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
001134' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001140'                         DI5:
                                        ;TEST THAT BIT2 OF MNCDI STATUS REGISTER IS READ-WRITE
001140' 012767  000004  176734          MOV     #BIT2,ACSR              ;LOAD EXPECTED
001146' 016777  176730  177064          MOV     ACSR,@ICSR              ;LOAD BIT2 INTO MNCDI STATUS REGISTER
001154' 017767  177060  176722          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
001162' 042767  000200  176714          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
001170' 026767  176706  176706          CMP     ACSR,ASTAT              ;TEST THAT IT SET
001176' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
001200' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT2 OF MNCDI STATUS REGISTER FAILED TO SET
                                        ;*************************************************************
001206' 046777  176670  177024  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
001214' 017767  177020  176662          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
001222' 026767  176654  176654          CMP     ACSR,ASTAT              ;TEST THE BIT
001230' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*************************************************************
001232' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT2 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                        ;*************************************************************
001240'                         2$:
001240' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
001244' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```



```
001250'                         DI6:
                                        ;TEST THAT BIT3 OF MNCDI STATUS REGISTER IS READ-WRITE
001250' 012767  000010  176624          MOV     #BIT3,ACSR              ;LOAD EXPECTED
001256' 016777  176620  176754          MOV     ACSR,@ICSR              ;LOAD BIT3 INTO MNCDI STATUS REGISTER
001264' 017767  176750  176612          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
001272' 042767  000200  176604          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
001300' 026767  176576  176576          CMP     ACSR,ASTAT              ;TEST THAT IT SET
001306' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
001310' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT3 OF MNCDI STATUS REGISTER FAILED TO SET
                                        ;*************************************************************
001316' 046777  176560  176714  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
001324' 017767  176710  176552          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
001332' 026767  176544  176544          CMP     ACSR,ASTAT              ;TEST THE BIT
001340' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*************************************************************
001342' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT3 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                        ;*************************************************************
001350'                         2$:
001350' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
001354' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001360'                         DI7:
                                        ;TEST THAT BIT4 OF MNCDI STATUS REGISTER IS READ-WRITE
001360' 012767  000020  176514          MOV     #BIT4,ACSR              ;LOAD EXPECTED
001366' 016777  176510  176644          MOV     ACSR,@ICSR              ;LOAD BIT4 INTO MNCDI STATUS REGISTER
001374' 017767  176640  176502          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
001402' 042767  000200  176474          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
001410' 026767  176466  176466          CMP     ACSR,ASTAT              ;TEST THAT IT SET
001416' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
001420' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT4 OF MNCDI STATUS REGISTER FAILED TO SET
                                        ;*************************************************************
001426' 046777  176450  176604  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
001434' 017767  176600  176442          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
001442' 026767  176434  176434          CMP     ACSR,ASTAT              ;TEST THE BIT
001450' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*************************************************************
001452' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT4 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                        ;*************************************************************
001460'                         2$:
001460' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
001464' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```

```
001470'                         DI10:
                                ;TEST THAT BIT5 OF MNCDI STATUS REGISTER IS READ-WRITE
001470' 012767  000040  176404          MOV     #BIT5,ACSR              ;LOAD EXPECTED
001476' 016777  176400  176534          MOV     ACSR,@ICSR              ;LOAD BIT5 INTO MNCDI STATUS REGISTER
001504' 017767  176530  176372          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
001512' 042767  000200  176364          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
001520' 026767  176356  176356          CMP     ACSR,ASTAT              ;TEST THAT IT SET
001526' 001403                          BEQ     1$                      ;BR IF SAME
                                ;*****************************************************************
001530' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT5 OF MNCDI STATUS REGISTER FAILED TO SET
                                ;*****************************************************************
001536' 046777  176340  176474  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
001544' 017767  176470  176332          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
001552' 026767  176324  176324          CMP     ACSR,ASTAT              ;TEST THE BIT
001560' 001003                          BNE     2$                      ;BR IF CLEARED
                                ;*****************************************************************
001562' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT5 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                ;*****************************************************************
001570'                         2$:
001570' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
001574' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001600'                         DI11:
                                ;TEST THAT BIT6 OF MNCDI STATUS REGISTER IS READ-WRITE
001600' 012767  000100  176274          MOV     #BIT6,ACSR              ;LOAD EXPECTED
001606' 016777  176270  176424          MOV     ACSR,@ICSR              ;LOAD BIT6 INTO MNCDI STATUS REGISTER
001614' 017767  176420  176262          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
001622' 042767  000200  176254          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
001630' 026767  176246  176246          CMP     ACSR,ASTAT              ;TEST THAT IT SET
001636' 001403                          BEQ     1$                      ;BR IF SAME
                                ;*****************************************************************
001640' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT6 OF MNCDI STATUS REGISTER FAILED TO SET
                                ;*****************************************************************
001646' 046777  176230  176364  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
001654' 017767  176360  176222          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
001662' 026767  176214  176214          CMP     ACSR,ASTAT              ;TEST THE BIT
001670' 001003                          BNE     2$                      ;BR IF CLEARED
                                ;*****************************************************************
001672' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT6 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                ;*****************************************************************
001700'                         2$:
001700' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
001704' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```



```
001710'                         DI12:
                                ;TEST THAT BIT8 OF MNCDI STATUS REGISTER IS READ-WRITE
001710' 012767  000400  176164          MOV     #BIT8,ACSR              ;LOAD EXPECTED
001716' 016777  176160  176314          MOV     ACSR,@ICSR              ;LOAD BIT8 INTO MNCDI STATUS REGISTER
001724' 017767  176310  176152          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
001732' 042767  000200  176144          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
001740' 026767  176136  176136          CMP     ACSR,ASTAT              ;TEST THAT IT SET
001746' 001403                          BEQ     1$                      ;BR IF SAME
                                ;*****************************************************************
001750' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT8 OF MNCDI STATUS REGISTER FAILED TO SET
                                ;*****************************************************************
001756' 046777  176120  176254  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
001764' 017767  176250  176112          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
001772' 026767  176104  176104          CMP     ACSR,ASTAT              ;TEST THE BIT
002000' 001003                          BNE     2$                      ;BR IF CLEARED
                                ;*****************************************************************
002002' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT8 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                ;*****************************************************************
002010'                         2$:
002010' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
002014' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002020'                         DI13:
                                ;TEST THAT BIT9 OF MNCDI STATUS REGISTER IS READ-WRITE
002020' 012767  001000  176054          MOV     #BIT9,ACSR              ;LOAD EXPECTED
002026' 016777  176050  176204          MOV     ACSR,@ICSR              ;LOAD BIT9 INTO MNCDI STATUS REGISTER
002034' 017767  176200  176042          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
002042' 042767  000200  176034          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
002050' 026767  176026  176026          CMP     ACSR,ASTAT              ;TEST THAT IT SET
002056' 001403                          BEQ     1$                      ;BR IF SAME
                                ;*****************************************************************
002060' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT9 OF MNCDI STATUS REGISTER FAILED TO SET
                                ;*****************************************************************
002066' 046777  176010  176144  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
002074' 017767  176140  176002          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
002102' 026767  175774  175774          CMP     ACSR,ASTAT              ;TEST THE BIT
002110' 001003                          BNE     2$                      ;BR IF CLEARED
                                ;*****************************************************************
002112' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT9 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                ;*****************************************************************
002120'                         2$:
002120' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
002124' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```

```
                                        ;TEST THAT BIT12 OF MNCDI STATUS REGISTER IS READ-WRITE
002130' 012767  010000  175744          MOV     #BIT12,ACSR             ;LOAD EXPECTED
002136' 016777  175740  176074          MOV     ACSR,@ICSR              ;LOAD BIT12 INTO MNCDI STATUS REGISTER
002144' 017767  176070  175732          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
002152' 042767  000200  175724          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
002160' 026767  175716  175716          CMP     ACSR,ASTAT              ;TEST THAT IT SET
002166' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*****************************************************************
002170' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT12 OF MNCDI STATUS REGISTER FAILED TO SET
                                        ;*****************************************************************
002176' 046777  175700  176034  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
002204' 017767  176030  175672          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
002212' 026767  175664  175664          CMP     ACSR,ASTAT              ;TEST THE BIT
002220' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*****************************************************************
002222' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT12 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                        ;*****************************************************************
002230'                         2$:
002230' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
002234' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002240'                         DI15:
                                        ;TEST THAT BIT14 OF MNCDI STATUS REGISTER IS READ-WRITE
002240' 012767  040000  175634          MOV     #BIT14,ACSR             ;LOAD EXPECTED
002246' 016777  175630  175764          MOV     ACSR,@ICSR              ;LOAD BIT14 INTO MNCDI STATUS REGISTER
002254' 017767  175760  175622          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER
002262' 042767  000200  175614          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
002270' 026767  175606  175606          CMP     ACSR,ASTAT              ;TEST THAT IT SET
002276' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*****************************************************************
002300' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT14 OF MNCDI STATUS REGISTER FAILED TO SET
                                        ;*****************************************************************
002306' 046777  175570  175724  1$:     BIC     ACSR,@ICSR              ;CLEAR THAT BIT
002314' 017767  175720  175562          MOV     @ICSR,ASTAT             ;READ MNCDI STATUS REGISTER AGAIN
002322' 026767  175554  175554          CMP     ACSR,ASTAT              ;TEST THE BIT
002330' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*****************************************************************
002332' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT14 OF MNCDI STATUS REGISTER FAILED TO CLEAR
                                        ;*****************************************************************
002340'                         2$:
002340' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
002344' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```



```
                                        ;VERIFY HIGH BYTE OPERATION ON THE INPUT STATUS REGISTER
002350' 012777  040424  175662  DI16:   MOV     #40424,@ICSR            ;LOAD INPUT REG. BIT
002356' 105077  175660                  CLRB    @ICSR1                  ;CLEAR HIGH BYTE
002362' 012767  000024  175512          MOV     #BIT4!BIT2,ACSR         ;LOAD EXPECTED
002370' 017767  175644  175506          MOV     @ICSR,ASTAT             ;READ INPUT STATUS REG.
002376' 042767  000200  175500          BIC     #BIT7,ASTAT             ;REMOVE BIT 7
002404' 026767  175472  175472          CMP     ACSR,ASTAT              ;COMPARE
002412' 001403                          BEQ     DI17
                                        ;*****************************************************************
002414' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;CLEARING HIGH BYTE CHANGED LOW BYTE
                                        ;*****************************************************************


                                        ;VERIFY LOW BYTE OPERATION ON THE INPUT STATUS REGISTER
002422' 012777  040426  175610  DI17:   MOV     #40426,@ICSR            ;LOAD INPUT REG.
002430' 105077  175604                  CLRB    @ICSR                   ;CLEAR LOW BYTE
002434' 012767  040400  175440          MOV     #40400,ACSR             ;LOAD EXPECTED
002442' 017767  175572  175434          MOV     @ICSR,ASTAT             ;READ INPUT STATUS REG.
002450' 026767  175426  175426          CMP     ACSR,ASTAT              ;COMPARE
002456' 001403                          BEQ     DI20                    ;BR IF SAME
                                        ;*****************************************************************
002460' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;CLEARING LOW BYTE CHANGED HIGH BYTE
                                        ;*****************************************************************


                                        ;VERIFY THAT MAINT. STROBE SETS "INPUT DATA READY"
002466' 005077  175546          DI20:   CLR     @ICSR                   ;ENSURE CLEAR FLAG
002472' 012767  000200  175402          MOV     #BIT7,ACSR              ;LOAD EXPECTED DATA
002500' 012777  004200  175532          MOV     #BITEXT!BIT7,@ICSR      ;GENERATE MAINT. STROBE
002506' 017767  175526  175370          MOV     @ICSR,ASTAT             ;READ INPUT STATUS REGISTER
002514' 026767  175362  175362          CMP     ACSR,ASTAT              ;COMPARE RESULTS
002522' 001403                          BEQ     DI21                    ;BR IF SAME
                                        ;*****************************************************************
002524' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;MAINT. STROBE FAILED TO SET "INPUT DATA READY"
                                        ;*****************************************************************
```

```
                                        ;VERIFY THAT "INPUT DATA READY" CAN BE WRITTEN TO A ZERO
002532' 005067  175344          DI21:   CLR     ACSR                    ;LOAD EXPECTED DATA
002536' 012777  004000  175474          MOV     #BITEXT,@ICSR           ;GENERATE MAINT. STROBE
002544' 005077  175470                  CLR     @ICSR                   ;CLEAR DATA READY FLAG
002550' 017767  175464  175326          MOV     @ICSR,ASTAT             ;READ INPUT STATUS REGISTER
002556' 026767  175320  175320          CMP     ACSR,ASTAT              ;COMPARE
002564' 001403                          BEQ     DI22                    ;BR IF SAME
                                        ;*************************************************************
002566' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;"INPUT DATA READY" FAILED TO BE WRITTEN TO A ZERO
                                        ;*************************************************************


                                        ;VERIFY THAT "INPUT DATA READY" WILL NOT SET IF IN STIMILUS MODE AND NO SBR MATC
002574' 005077  175450          DI22:   CLR     @SBR                    ;CLEAR SBR REGISTER
002600' 012777  177777  175436          MOV     #-1,@DIR                ;CLEAR INPUT REGISTER
002606' 012777  000004  175424          MOV     #BIT2,@ICSR             ;SET STILILUS MODE
002614' 052777  004000  175416          BIS     #BITEXT,@ICSR           ;GENERATE MAINT. STROBE
002622' 012767  000004  175252          MOV     #BIT2,ACSR              ;LOAD EXPECTED
002630' 017767  175404  175246          MOV     @ICSR,ASTAT             ;READ STATUS
002636' 026767  175240  175240          CMP     ACSR,ASTAT              ;COMPARE
002644' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
002646' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INPUT STROBE SET INPUT READY WHEN IN STIMILUS MODE
                                        ;*************************************************************
002654'                         1$:
002654' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
002660' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                        ;VERIFY THAT "OVERRUN ERROR" SETS
002664' 012767  100202  175210  DI23:   MOV     #BIT15!BIT7!BIT1,ACSR   ;LOAD EXPECTED
002672' 012777  000002  175340          MOV     #BIT1,@ICSR             ;SET STROBE MODE
002700' 052777  004200  175332          BIS     #BITEXT!BIT7,@ICSR      ;GENERATE MAINT, STROBE
002706' 052777  104200  175324          BIS     #BIT15!BITEXT!BIT7,@ICSR        ;GENERATE MAINT. STROBE AGAIN
002714' 017767  175320  175162          MOV     @ICSR,ASTAT             ;READ INPUT STATUS REGISTER
002722' 026767  175154  175154          CMP     ACSR,ASTAT              ;COMPARE
002730' 001403                          BEQ     DI24                    ;BR IF SAME
                                        ;*************************************************************
002732' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;"OVER RUN" FAILED TO SET
                                        ;*************************************************************


                                        ;VERIFY THAT "OVERRUN ERROR" CAN BE WRITTEN TO A ZERO
002740' 012767  000202  175134  DI24:   MOV     #BIT7!BIT1,ACSR ;LOAD EXPECTED VALUE
002746' 012777  000002  175264          MOV     #BIT1,@ICSR             ;SET STROBE MODE
002754' 052777  004200  175256          BIS     #BITEXT!BIT7,@ICSR      ;GENERATE MAINT. STROBE
002762' 052777  104200  175250          BIS     #BIT15!BITEXT!BIT7,@ICSR        ;GENERATE MAINT. STROBE AGAN
002770' 105077  175246                  CLRB    @ICSR1                  ;CLEAR HIGH BYTE OF THE INPUT STATUS REGISTER
002774' 017767  175240  175102          MOV     @ICSR,ASTAT             ;READ INPUT STATUS REGISTER
003002' 026767  175074  175074          CMP     ACSR,ASTAT              ;COMPARE
003010' 001403                          BEQ     DI25                    ;BR IF SAME
                                        ;*************************************************************
003012' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;"OVERRUN ERROR" FAILED TO BE WRITTEN TO A ZERO
                                        ;*************************************************************
```



```
                                        ;VERIFY INVERT DATA FUNCTION
003020' 005067  175056          DI25:   CLR     ACSR                    ;LOADEXPECTED
003024' 012777  010000  175206          MOV     #BITDAT,@ICSR           ;SET INPUT INHIBIT
003032' 017767  175206  175044          MOV     @DIR,ASTAT              ;READ INPUT
003040' 026767  175036  175036          CMP     ACSR,ASTAT              ;COMPARE
003046' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
003050' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INPUT INHIBIT FAILED TO INHIBIT INPUT
                                        ;*************************************************************
003056' 012777  010060  175154  1$:     MOV     #BITDAT!BIT5!BIT4,@ICSR ;SET INVERT DATA AND INPUT INHIBIT
003064' 012767  177777  175010          MOV     #-1,ACSR                ;LOAD EXPECTED
003072' 017767  175146  175004          MOV     @DIR,ASTAT              ;READ INPUT
003100' 001003                          BNE     2$                      ;BR IF NON-ZERO
                                        ;*************************************************************
003102' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INVERT DATA FUNCTION FAILED
                                        ;*************************************************************
003110' 026767  174766  174766  2$:     CMP     ACSR,ASTAT              ;COMPARE DATA
003116' 001403                          BEQ     3$                      ;BR IF SAME
                                        ;*************************************************************
003120' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INVERT DATA - DATA PATH ERROR
                                        ;*************************************************************


003126' 012777  010000  175104  3$:     MOV     #BITDAT,@ICSR           ;SET INPUT INHIBIT
003134' 005067  174742                  CLR     ACSR                    ;CLEAR EXPECTED
003140' 017767  175100  174736          MOV     @DIR,ASTAT              ;READ INPUT
003146' 026767  174730  174730          CMP     ACSR,ASTAT              ;COMPARE
003154' 001403                          BEQ     DI26                    ;BR IF SAME
                                        ;*************************************************************
003156' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INVERT DATA FUNCTION OR INPUT INHIBIT FAILED
                                        ;*************************************************************


                                        ;VERIFY EACH BIT OF THE MNCDI INPUT DATA REGISTER CAN BE CLEARED
003164' 012767  000001  175100  DI26:   MOV     #BIT0,TEMP              ;LOAD INITIAL BIT
003172' 012777  010060  175040  1$:     MOV     #BITDAT!BIT4!BIT5,@ICSR ;LOAD INHIBIT INPUT AND INVERT DATA
003200' 016767  175066  174674          MOV     TEMP,ACSR               ;LOAD EXPECTED
003206' 005167  174670                  COM     ACSR                    ;MAKE OPPOSITE
003212' 017700  175026                  MOV     @DIR,R0                 ;READ INPUT
003216' 016777  175050  175020          MOV     TEMP,@DIR               ;CLEAR THE INPUT BIT
003224' 042777  000040  175006          BIC     #BIT5,@ICSR             ;REM INVERT DATA BITOVE
003232' 052777  000002  175000          BIS     #BIT1,@ICSR             ;ENABLE EXT. STROBE TO PREVENT  DATA INPUT BEING
003240' 017767  175000  174636          MOV     @DIR,ASTAT              ;READ INPUT REG.
003246' 026767  174630  174630          CMP     ACSR,ASTAT              ;COMPARE
003254' 001403                          BEQ     2$                      ;BR IF SAME
                                        ;*************************************************************
003256' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INPUT REGISTER BIT FAILED TO CLEAR
                                        ;*************************************************************
003264'                         2$:
003264' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
003270' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003274' 006367  174772                  ASL     TEMP                    ;SHIFT THE DATA
003300' 001334                          BNE     1$                      ;TRY MORE BITS
```

```
                                        ;VERIFY THAT A 2ND STROBE PULSE WILL NOT CHANGE THE DIR DATA
003302' 012777  010062  174730  DI27:   MOV     #BIT12!BIT5!BIT4!BIT1,@ICSR     ;DISABLE INPUTS, ENABLE INVERT DATA, EXT
003310' 052777  004200  174722          BIS     #BITEXT!BIT7,@ICSR      ;GENERATE MAINT. STROBE
003316' 042777  000040  174714          BIC     #BIT5,@ICSR             ;REMOVE INVERT DATA
003324' 052777  004200  174706          BIS     #BITEXT!BIT7,@ICSR      ;SET MAINT. STROBE AGAIN
003332' 012767  177777  174542          MOV     #-1,ACSR                ;LOAD EXPECTED DATA
003340' 017767  174700  174536          MOV     @DIR,ASTAT              ;READ REGISTER
003346' 026767  174530  174530          CMP     ACSR,ASTAT              ;COMPARE
003354' 001403                          BEQ     DI30                    ;BR IF SAME
                                        ;*************************************************************
003356' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;DATA READY FAILED TO INHIBIT 2ND
                                        ;*************************************************************
                                                                ;STROBE FROM CHAINING THE DIR
                                        ;INTERRUPT TEST -- VERIFY MNCDI INTERRUPTS VIA DATA READY VECTOR
003364' 012777  003430' 174666  DI30:   MOV     #1$,@DIDINV             ;LOAD RETURN VECTOR
003372' 116777  174414  174662          MOVB    BR1,@DIDIN$             ;LOAD RETURN LEVEL
003400' 012777  000102  174632          MOV     #BIT6!BIT1,@ICSR        ;SET STROBE MODE
003406' 052777  004200  174624          BIS     #BITEXT!BIT7,@ICSR      ;GENERATE MAINT. STROBE
003414' 000240                          NOP
003416' 000240                          NOP
003420' 000240                          NOP
003422' 000240                          NOP
003424' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.

003430' 005077  174604          1$:     CLR     @ICSR
                                        ;-------------------------------------------------------------
003434' 000004  000000' 003442'         PIRQ$,BEGIN,2$          ; QUEUE UP TO CONTINUE AT 2$ AND RTI
                                        ;-------------------------------------------------------------
003442' 012777  000262' 174610  2$:     MOV     #DIDIN$,@DIDINV
003450' 005077  174606                  CLR     @DIDIN$


                                        ;INTERRUPT TEST -- VERIFY MNCDI INTERRUPTS VIA OVERRUN ERROR
003454' 012777  003524' 174602  DI31:   MOV     #1$,@DIEINV             ;LOAD RETURN VECTOR
003462' 116777  174324  174576          MOVB    BR1,@DIEIN$             ;LOAD RETURN STATUS
003470' 012777  040002  174542          MOV     #BIT14!BIT1,@ICSR       ;ENABLE INTR. AND STROBE MODE
003476' 052777  104200  174534          BIS     #BIT15!BITEXT!BIT7,@ICSR        ;GENERATE MAINT. STROBE
003504' 052777  104200  174526          BIS     #BIT15!BITEXT!BIT7,@ICSR        ;GENERATE MAINT. STROBE AGAIN TO SET OVE
003512' 000240                          NOP
003514' 000240                          NOP
003516' 000240                          NOP
003520' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
003524' 005077  174510          1$:     CLR     @ICSR
                                        ;-------------------------------------------------------------
003530' 000004  000000' 003536'         PIRQ$,BEGIN,2$          ; QUEUE UP TO CONTINUE AT 2$ AND RTI
                                        ;-------------------------------------------------------------
003536' 012777  000266' 174520  2$:     MOV     #DIEIN$,@DIEINV
003544' 005077  174516                  CLR     @DIEINS
003550' 036767  174522  174512          BIT     TEMP2,SRMINE            ;TEST IF INPUT ONLY
003556' 001002                          BNE     DO0                     ;BR IF WRAP-AROUND MODE
003560' 000167  174634                  JMP     CONT2                   ;NO -- TRY NEXT UNIT
```



```
                                        ;VERIFY CORRECT MNCDO ADDRESS RESPONSE
003564' 016767  174440  174306  DO0:    MOV     OCSR,CSRA               ;LOAD BUS ADDRESS
003572' 005777  174432                  TST     @OCSR                   ;TEST OUTPUT STATUS REGISTER
003576' 005777  174432                  TST     @DOR                    ;TEST OUTPUT DATA REGISTER
003602'                         DO1:
                                        ;FLOAT A 1 ACROSS THE MNCDO DATA REGISTER
003602' 012767  000001  174272          MOV     #BIT0,ACSR              ;LOAD EXPECT BIT
003610' 016777  174266  174416  1$:     MOV     ACSR,@DOR               ;LOAD MNCDO DATA REGISTER
003616' 017767  174412  174260          MOV     @DOR,ASTAT              ;READ MNCDO DATA REGISTER
003624' 026767  174252  174252          CMP     ACSR,ASTAT              ;COMPARE
003632' 001403                          BEQ     2$                      ;BR IF SAME
                                        ;*************************************************************
003634' 104405  000000' 000000          HRDER$,BEGIN,NULL       ; MNCDO DATA REGISTER FAILED TO HOLD A FLOATING 1
                                        ;*************************************************************
003642'                         2$:
003642' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
003646' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003652' 006367  174224                  ASL     ACSR                    ;CHANGE THE DATA
003656' 001354                          BNE     1$                      ;BR IF MORE DATA
003660'                         DO2:
                                        ;FLOAT A 0 ACROSS THE MNCDO DATA REGISTER
003660' 012767  000001  174404          MOV     #BIT0,TEMP              ;LOAD INITIAL BIT
003666' 016767  174400  174206  1$:     MOV     TEMP,ACSR               ;LOAD EXPECTED
003674' 005167  174202                  COM     ACSR                    ;COMPLEMENT
003700' 016777  174176  174326          MOV     ACSR,@DOR               ;LOAD MNCDO DATA REGISTER
003706' 017767  174322  174170          MOV     @DOR,ASTAT              ;READ MNCDO DATA REGISTER
003714' 026767  174162  174162          CMP     ACSR,ASTAT              ;COMPARE
003722' 001403                          BEQ     2$                      ;BR IF SAME
                                        ;*************************************************************
003724' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;MNCDO DATA REGISTER FAILED TO HOLD A FLOATING 0
                                        ;*************************************************************
003732'                         2$:
003732' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
003736' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003742' 006367  174324                  ASL     TEMP                    ;CHANGE THE DATA
003746' 001347                          BNE     1$                      ;BR IF MORE DATA
003750'                         DO3:
                                        ;VERIFY BYTE OPERATION ON THE MNCDO DATA REGISTER
003750' 012777  177777  174256  1$:     MOV     #-1,@DOR                ;LOAD MNCDO DATA REGISTER
003756' 012767  000377  174116          MOV     #377,ACSR               ;LOAD EXPECTED
003764' 105077  174246                  CLRB    @DOR1                   ;CLEAR HIGH BYTE
003770' 017767  174240  174106          MOV     @DOR,ASTAT              ;READ MNCDO DATA REGISTER
003776' 026767  174100  174100          CMP     ACSR,ASTAT              ;COMPARE
004004' 001403                          BEQ     2$                      ;BR IF SAME
                                        ;*************************************************************
004006' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;CLEARING HIGH BYTE CHANGED LOW BYTE
                                        ;*************************************************************
004014' 012777  177777  174212  2$:     MOV     #-1,@DOR                ;LOAD MNCDO DATA REGISTER
004022' 012767  177400  174052          MOV     #177400,ACSR            ;LOAD EXPECTED
004030' 105077  174200                  CLRB    @DOR                    ;CLEAR LOW BYTE
004034' 017767  174174  174042          MOV     @DOR,ASTAT              ;READ MNCDO DATA REGISTER
004042' 026767  174034  174034          CMP     ACSR,ASTAT              ;COMPARE
004050' 001403                          BEQ     3$                      ;BR IF SAME
                                        ;*************************************************************
004052' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;CLEARING LOW BYTE CHANGED HIGH BYTE
                                        ;*************************************************************
```

```
                                        3$:
004060' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
004064' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
004070'                                 DO4:
                                        ;TEST THAT BIT6 OF MNCDO STATUS REGISTER IS READ-WRITE
004070' 012767  000100  174004          MOV     #BIT6,ACSR              ;LOAD EXPECTED
004076' 016777  174000  174124          MOV     ACSR,@OCSR              ;LOAD BIT6 INTO MNCDO STATUS REGISTER
004104' 017767  174120  173772          MOV     @OCSR,ASTAT             ;READ MNCDO STATUS REGISTER
004112' 042767  000200  173764          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
004120' 026767  173756  173756          CMP     ACSR,ASTAT              ;TEST THAT IT SET
004126' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
004130' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT6 OF MNCDO STATUS REGISTER FAILED TO SET
                                        ;*************************************************************
004136' 046777  173740  174064  1$:     BIC     ACSR,@OCSR              ;CLEAR THAT BIT
004144' 017767  174060  173732          MOV     @OCSR,ASTAT             ;READ MNCDO STATUS REGISTER AGAIN
004152' 026767  173724  173724          CMP     ACSR,ASTAT              ;TEST THE BIT
004160' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*************************************************************
004162' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT6 OF MNCDO STATUS REGISTER FAILED TO CLEAR
                                        ;*************************************************************
004170'                                 2$:
004170' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
004174' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
004200'                                 DO5:
                                        ;TEST THAT BIT4 OF MNCDO STATUS REGISTER IS READ-WRITE
004200' 012767  000020  173674          MOV     #BIT4,ACSR              ;LOAD EXPECTED
004206' 016777  173670  174014          MOV     ACSR,@OCSR              ;LOAD BIT4 INTO MNCDO STATUS REGISTER
004214' 017767  174010  173662          MOV     @OCSR,ASTAT             ;READ MNCDO STATUS REGISTER
004222' 042767  000200  173654          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
004230' 026767  173646  173646          CMP     ACSR,ASTAT              ;TEST THAT IT SET
004236' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
004240' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT4 OF MNCDO STATUS REGISTER FAILED TO SET
                                        ;*************************************************************
004246' 046777  173630  173754  1$:     BIC     ACSR,@OCSR              ;CLEAR THAT BIT
004254' 017767  173750  173622          MOV     @OCSR,ASTAT             ;READ MNCDO STATUS REGISTER AGAIN
004262' 026767  173614  173614          CMP     ACSR,ASTAT              ;TEST THE BIT
004270' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*************************************************************
004272' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT4 OF MNCDO STATUS REGISTER FAILED TO CLEAR
                                        ;*************************************************************
004300'                                 2$:
004300' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
004304' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```



```
004310'                                 DO6:
                                        ;TEST THAT BIT3 OF MNCDO STATUS REGISTER IS READ-WRITE
004310' 012767  000010  173564          MOV     #BIT3,ACSR              ;LOAD EXPECTED
004316' 016777  173560  173704          MOV     ACSR,@OCSR              ;LOAD BIT3 INTO MNCDO STATUS REGISTER
004324' 017767  173700  173552          MOV     @OCSR,ASTAT             ;READ MNCDO STATUS REGISTER
004332' 042767  000200  173544          BIC     #BIT7,ASTAT             ;CLEAR BIT 7
004340' 026767  173536  173536          CMP     ACSR,ASTAT              ;TEST THAT IT SET
004346' 001403                          BEQ     1$                      ;BR IF SAME
                                        ;*************************************************************
004350' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT3 OF MNCDO STATUS REGISTER FAILED TO SET
                                        ;*************************************************************
004356' 046777  173520  173644  1$:     BIC     ACSR,@OCSR              ;CLEAR THAT BIT
004364' 017767  173640  173512          MOV     @OCSR,ASTAT             ;READ MNCDO STATUS REGISTER AGAIN
004372' 026767  173504  173504          CMP     ACSR,ASTAT              ;TEST THE BIT
004400' 001003                          BNE     2$                      ;BR IF CLEARED
                                        ;*************************************************************
004402' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BIT3 OF MNCDO STATUS REGISTER FAILED TO CLEAR
                                        ;*************************************************************
004410'                                 2$:
004410' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
004414' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                        ;VERIFY THAT MNCDO DONE FLAG SETS
004420' 005077  173604          DO7:    CLR     @OCSR                   ;CLEAR CLEARED FLAG
004424' 012767  000200  173450          MOV     #BIT7,ACSR              ;LOAD EXPECTED
004432' 105077  173576                  CLRB    @DOR                    ;ENABLE
004436' 112777  000001  173566          MOVB    #BIT0,@OCSR1            ;GENERATE MAINT. REPLY
004444' 017767  173560  173432          MOV     @OCSR,ASTAT             ;READ OUTPUT STATUS REGISTER
004452' 026767  173424  173424          CMP     ACSR,ASTAT              ;COMMPARE
004460' 001403                          BEQ     DO10                    ;BR IF SAME
                                        ;*************************************************************
004462' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;OUTPUT DONE FLAG FAILED TO SET
                                        ;*************************************************************
                                        ;VERIFY THAT MNCDO DONE FLAG CLEARS WHEN WRITTEN TO A 0
004470' 105077  173540          DO10:   CLRB    @DOR                    ;ENABLE
004474' 112777  000001  173530          MOVB    #BIT0,@OCSR1            ;GENERATE MAINT. REPLY
004502' 005067  173374                  CLR     ACSR                    ;CLEAR EXPECTED
004506' 005077  173516                  CLR     @OCSR                   ;CLEAR OUTPUT DONE FLAG
004512' 017767  173512  173364          MOV     @OCSR,ASTAT             ;READ STATUS
004520' 001403                          BEQ     DO11                    ;BR IF SAME
                                        ;*************************************************************
004522' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;CLEARING THE OUTPUT FLAG FAILED TO CLEAR OUTPUT DONE FL
                                        ;*************************************************************
                                        ;VERIFY THAT MNCDO DONE FLAG CLEARS WHEN OUTPUT DATA REGISTER IS WRITTEN
004530' 105077  173500          DO11:   CLRB    @DOR                    ;ENABLE
004534' 112777  000001  173470          MOVB    #BIT0,@OCSR1            ;GENERATE MAINT. REPLY
004542' 005067  173334                  CLR     ACSR                    ;CLEAR EXPECTED
004546' 005077  173462                  CLR     @DOR                    ;WRITE THE OUTPUT DATA REGISTER
004552' 017767  173452  173324          MOV     @OCSR,ASTAT             ;READ OUTPUT STATUS REGISTER
004560' 001403                          BEQ     DO12                    ;BR IF CLEARED
                                        ;*************************************************************
004562' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;OUTPUT DONE FLAG FAILED TO CLEAR WHEN THE DOR REGISTER
                                        ;*************************************************************
```

```
                                        ;INTERRUPT TEST -- VERIFY MNCDO DOES INTERRUPT
004570' 012777  004636' 173456  DO12:  MOV     #1$,@DODINV             ;LOAD RETURN VECTOR
004576' 116777  173210  173452          MOVB    BR1,@DODINS             ;LOAD RETURN LEVEL
004604' 105077  173424                  CLRB    @DOR                    ;ENABLE
004610' 112777  000001  173414          MOVB    #BIT0,@OCSR1            ;GENERATE MAINT. REPLY
004616' 052777  000100  173404          BIS     #BIT6,@OCSR             ;ENABLE INTR.
004624' 000240                          NOP
004626' 000240                          NOP
004630' 000240                          NOP
004632' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

004636' 005077  173366          1$:     CLR     @OCSR
                                        ;---------------------------------------------------------------
004642' 000004  000000' 004650'         PIRQ$,BEGIN,3$          ; QUEUE UP TO CONTINUE AT 3$ AND RTI
                                        ;---------------------------------------------------------------
004650' 016777  173402  173376  3$:     MOV     DODINS,@DODINV          ;RESET VECTORS
004656' 005077  173374                  CLR     @DODINS
004662' 005077  173342                  CLR     @OCSR
```



```
                                ;WRAP AROUND TESTS

                                        ;VERFIY MODE 00  -- INPUT STROBE WILL SET THE INPUT DATA READY FLAG
004666' 012777  177777  173350  DI32:  MOV     #-1,@DIR                ;CLEAR INPUT REG.
004674' 005077  173350                  CLR     @SBR                    ;CLEAR STIM. REG.
004700' 005077  173334                  CLR     @ICSR                   ;CLEAR INPUT DATA READY FLAG
004704' 005077  173320                  CLR     @OCSR                   ;CLEAR OUTPUT STATUS
004710' 012777  025252  173316          MOV     #25252,@DOR             ;WRITE TO THE OUTPUT DATA REG.
004716' 012767  000200  173156          MOV     #BIT7,ACSR              ;LOAD EXPECTED
004724' 017767  173310  173152          MOV     @ICSR,ASTAT             ;READ STATUS
004732' 026767  173144  173144          CMP     ACSR,ASTAT              ;COMPARE
004740' 001406                          BEQ     DI33                    ;BR IF SAME
                                        ;*************************************************************
004742' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;MODE 00  -- EXT. STROBE FAILED TO SET INPUT DATA READY
                                        ;*************************************************************
004750' 104403  000000' 006030'         MSGN$,BEGIN,MSG1        ;ASCII MESSAGE CALL WITH COMMON HEADER

                                        ;VERIFY MODE 01  -- INPUT STROBE WILL SET THE INPUT DATA READY FLAG
004756' 012777  177777  173260  DI33:  MOV     #-1,@DIR                ;CLEAR INPUT REG.
004764' 005077  173260                  CLR     @SBR                    ;CLEAR STIM. REG.
004770' 012777  000002  173242          MOV     #BIT1,@ICSR             ;CLEAR INPUT DATA READY FLAG
004776' 005077  173226                  CLR     @OCSR                   ;CLEAR OUTPUT STATUS
005002' 012777  052525  173224          MOV     #52525,@DOR             ;WRITE TO THE OUTPUT DATA REG.
005010' 012767  000202  173064          MOV     #BIT7!BIT1,ACSR ;LOAD EXPECTED
005016' 017767  173216  173060          MOV     @ICSR,ASTAT             ;READ STATUS
005024' 026767  173052  173052          CMP     ACSR,ASTAT              ;COMPARE
005032' 001403                          BEQ     DI34                    ;BR IF SAME
                                        ;*************************************************************
005034' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;MODE 01  -- EXT. STROBE FAILED TO SET INPUT DATA READY
                                        ;*************************************************************

                                        ;VERIFY MODE 10  -- INPUT STROBE WILL NOT SET INPUT DATA READY FLAG
005042' 012777  177777  173174  DI34:  MOV     #-1,@DIR                ;CLEAR INPUT REG.
005050' 005077  173174                  CLR     @SBR                    ;CLEAR STIM. REG.
005054' 012777  000004  173156          MOV     #BIT2,@ICSR             ;CLEAR INPUT DATA READY FLAG
005062' 005077  173142                  CLR     @OCSR                   ;CLEAR OUTPUT STATUS
005066' 012777  070707  173140          MOV     #70707,@DOR             ;WRITE TO THE OUTPUT DATA REG.
005074' 012767  000004  173000          MOV     #BIT2,ACSR              ;LOAD EXPECTED
005102' 017767  173132  172774          MOV     @ICSR,ASTAT             ;READ STATUS
005110' 026767  172766  172766          CMP     ACSR,ASTAT              ;COMPARE
005116' 001403                          BEQ     DI35                    ;BR IF SAME
                                        ;*************************************************************
005120' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;MODE 10  -- EXT. STROBE SET INPUT DATA READY FLAG IN ER
                                        ;*************************************************************
```

```
                                        ;VERIFY INPUT REPLY SETS OUTPUT DONE FLAG
005126' 005077  173102          DI35:   CLR     @DOR                    ;CLEAR OUTPUT DATA
005132' 012767  000200  172742          MOV     #BIT7,ACSR              ;LOAD EXPECTED
005140' 005077  173064                  CLR     @OCSR                   ;CLEAR OUTPUT DONE FLAG
005144' 012777  004200  173066          MOV     #BITEXT!BIT7,@ICSR      ;SET INPUT READY FLAG
005152' 005077  173062                  CLR     @ICSR                   ;CLEAR INPUT READY FLAG<GEN, INPUT REPLY>
005156' 017767  173046  172720          MOV     @OCSR,ASTAT             ;READ OUTPUT STATUS
005164' 026767  172712  172712          CMP     ACSR,ASTAT              ;COMPARE
005172' 001403                          BEQ     DI36                    ;BR IF SAME
                                        ;*************************************************************
005174' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INPUT REPLY FAILED TO SET OUTPUT DONE FLAG
                                        ;*************************************************************
                                        ;VERIFY THE MNCDO - WRAPAROUND - MNCDI DATA PATH
005202' 012777  177777  173034  DI36:   MOV     #-1,@DIR                ;CLEAR INPUT REG.
005210' 005077  173014                  CLR     @OCSR                   ;CLEAR OUTPUT STATUS
005214' 005077  173020                  CLR     @ICSR                   ;CLEAR INPUT STATUS
005220' 012767  000001  173044          MOV     #BIT0,TEMP              ;LOAD EXPECTED
005226' 016767  173040  172646  1$:     MOV     TEMP,ACSR               ;LOAD TYPEOUT EXPECTED
005234' 016777  172642  172772          MOV     ACSR,@DOR               ;LOAD OUTPUT DATA REG.
005242' 017767  172776  172634          MOV     @DIR,ASTAT              ;READ INPUT DATA REGISTER
005250' 026767  172626  172626          CMP     ACSR,ASTAT              ;COMPARE
005256' 001403                          BEQ     2$                      ;BR IF SAME
                                        ;*************************************************************
005260' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INPUT DATA PATH ERROR
                                        ;*************************************************************
005266'                         2$:
005266' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
005272' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
005276' 006367  172770                  ASL     TEMP                    ;TRY NEXT BIT
005302' 001351                          BNE     1$                      ;BR IF MORE BITS
                                        ;VERIFY THE MNCDO - WRAPAROUND - MNCDI INVERTED DATA PATH
005304' 012777  177777  172732  DI37:   MOV     #-1,@DIR                ;CLEAR INPUT REG.
005312' 005077  172712                  CLR     @OCSR                   ;CLEAR OUTPUT STATUS
005316' 012777  000060  172714          MOV     #BIT5!BIT4,@ICSR        ;LOAD INPUT STATUS <INVERT DATA>
005324' 012767  000001  172740          MOV     #BIT0,TEMP              ;LOAD INITIAL BIT
005332' 016777  172734  172674  1$:     MOV     TEMP,@DOR               ;LOAD OUTPUT DATA REG.
005340' 017767  172700  172536          MOV     @DIR,ASTAT              ;READ INPUT DATA REGISTER
005346' 016767  172720  172526          MOV     TEMP,ACSR               ;GET THE BIT
005354' 005167  172522                  COM     ACSR                    ;INVERT EXPECTED INPUT DATA
005360' 026767  172516  172516          CMP     ACSR,ASTAT              ;COMPARE
005366' 001403                          BEQ     2$                      ;BR IF SAME
                                        ;*************************************************************
005370' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INVERTED INPUT DATA PATH ERROR
                                        ;*************************************************************
005376'                         2$:
005376' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
005402' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
005406' 006367  172660                  ASL     TEMP                    ;TRY NEXT BIT
005412' 001347                          BNE     1$                      ;BR IF MORE BITS
```



```
                                        ;VERIFY IN 10 MODE THAT SBR AND INPUT  BITS SET INPUT READY
                                ;       INPUT DATA READY FLAG
                                ;               LOAD A FLOATING 1 ACROSS THE SBR
                                ;               VERIFY THAT ONLY THE CORRECT BIT SET DATA READY
005414' 005077  172614          DI40:   CLR     @DOR
005420' 012767  000001  172644          MOV     #BIT0,TEMP              ;LOAD INITIAL BIT
005426' 005077  172602          1$:     CLR     @DOR                    ;CLEAR OUTPUT BITS
005432' 005077  172612                  CLR     @SBR                    ;CLEAR SBR REG.
005436' 012777  000004  172574          MOV     #BIT2,@ICSR             ;CLEAR INPUT READY AND SET MODE 10
005444' 012777  177777  172572          MOV     #-1,@DIR                ;CLEAR INPUT REG.
005452' 016777  172614  172570          MOV     TEMP,@SBR               ;LOAD SBR REG.
005460' 042777  000200  172552          BIC     #BIT7,@ICSR             ;CLEAR INPUT READY BIT
005466' 016777  172600  172540          MOV     TEMP,@DOR               ;LOAD OUTPUT REG.
005474' 012767  100204  172400          MOV     #BIT15!BIT7!BIT2,ACSR   ;LOAD EXPECTED STATUS
005502' 017767  172532  172374          MOV     @ICSR,ASTAT             ;READ STATUS
005510' 026767  172366  172366          CMP     ACSR,ASTAT              ;COMPARE
005516' 001403                          BEQ     2$                      ;BR IF SAME
                                        ;*************************************************************
005520' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INPUT DATA READY FLAG FAILED
                                        ;*************************************************************
                                                                ;TO SET IN MODE 10 <STIMILUS MODE>
                                        ;NOW LOAD ALL BITS EXCEPT THE FLOATING BIT AND ENSURE INPUT DATA READY DOES NOT SET
005526' 016767  172540  172540  2$:     MOV     TEMP,TEMP1              ;COPY EXPECTED
005534' 005167  172534                  COM     TEMP1                   ;USE REVERSE PATTERN
005540' 005077  172470                  CLR     @DOR                    ;CLEAR OUTPUT REG.
005544' 012777  177777  172472          MOV     #-1,@DIR                ;CLEAR INPUT REG.
005552' 042777  100200  172460          BIC     #BIT15!BIT7,@ICSR       ;CLEAR INPUT DATA READY
005560' 012767  000004  172314          MOV     #BIT2,ACSR              ;LOAD EXPECTED
005566' 016777  172502  172440          MOV     TEMP1,@DOR              ;LOAD ALL OTHER DATA BITS
005574' 017767  172440  172302          MOV     @ICSR,ASTAT             ;READ INPUT STATUS
005602' 026767  172274  172274          CMP     ACSR,ASTAT              ;COMPARE
005610' 001403                          BEQ     3$                      ;BR IF SAME
                                        ;*************************************************************
005612' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INPUT DATA READY FLAG SET IN ERROR
                                        ;*************************************************************
                                                                ;UNEXPECTED SBR BIT SET INPUT DATA READY
005620'                         3$:
005620' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
005624' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
005630' 006367  172436                  ASL     TEMP                    ;TRY NEXT BIT
005634' 001274                          BNE     1$                      ;BR IF MORE BITS
```

```
                                        ;TEST THE TRANSITION ENABLE AND TRANSITION DETECTION
005636' 005077  172372          DI4$:   CLR     @DOR
005642' 012777  177777  172374          MOV     #-1,@DIR                ;CLEAR INPUT REGISTER
005650' 012767  010000  172414          MOV     #BIT12,TEMP             ;LOAD INITIAL TRANSITION BIT
005656' 012777  000404  172354          MOV     #BIT8!BIT2,@ICSR        ;SET STIM, CLEAR READY, ENABLE TRANS.
005664' 012767  100604  172210          MOV     #BIT15!BIT8!BIT7!BIT2,ACSR      ;LOAD EXPECTED STATUS
005672' 042777  100200  172340  1$:     BIC     #BIT15!BIT7,@ICSR       ;CLEAR READY
005700' 016777  172366  172342          MOV     TEMP,@SBR               ;LOAD STIMULUS REG.
005706' 016777  172360  172320          MOV     TEMP,@DOR               ;LOAD INPUT REG. <VIA OUTPUT REG.>
005714' 017767  172320  172162          MOV     @ICSR,ASTAT             ;READ INPUT STATUS
005722' 026767  172154  172154          CMP     ACSR,ASTAT              ;COMPARE
005730' 001403                          BEQ     2$                      ;BR IF SAME
                                        ;*************************************************************
005732' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;TRANSITION ENABLE OR TRANSITION TO A ONE FAILED
                                        ;*************************************************************

                                ;NOW REMOVE THE TRANSITION DATA BIT (THIS SHOULD CAUSE THE INPUT READY FLAG TO SET AGAIN
005740' 012777  177777  172276  2$:     MOV     #-1,@DIR                ;CLEAR INPUT REG
005746' 042777  100200  172264          BIC     #BIT15!BIT7,@ICSR       ;CLEAR INPUT READY FLAG
005754' 046777  172312  172252          BIC     TEMP,@DOR               ;REMOVE THE INPUT DATA
                                                                ;THIS SHOULD CAUSE THE TRANSITION TO A ZERO
005762' 017767  172252  172114          MOV     @ICSR,ASTAT             ;READ INPUT STATUS
005770' 026767  172106  172106          CMP     ACSR,ASTAT              ;COMPARE
005776' 001403                          BEQ     3$                      ;BR IF SAME
                                        ;*************************************************************
006000' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;TRANSITION TO A ZERO FAILED
                                        ;*************************************************************

006006'                         3$:
006006' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
006012' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
006016' 006367  172250                  ASL     TEMP                    ;TRY ANOTHER BIT ?
006022' 001323                          BNE     1$                      ;BR IF YES

006024' 000167  172370                  JMP     CONT2                   ;TRY NEXT UNIT

006030' 006034'                 MSG1:   PRI1                    ;POINTER TO TEXT
006032' 177777                          -1                      ;TERMINATOR
006034' 053045  051105  043111  PRI1:   .ASCIZ  \%VERIFY MNCDO WRAP-AROUND CONNECTION TO MNCDI\
006042' 020131  047115  042103
006050' 020117  051127  050101
006056' 040455  047522  047125
006064' 020104  047503  047116
006072' 041505  044524  047117
006100' 052040  020117  047115
006106' 042103  000111
                                        .EVEN
          000001                        .END
```



```
ACSR    000102R          198#   305*   306    308    316*   321*   322*   323    325    338*   341    347*   350
                         360*   361    364    369    371    381*   382    385    390    392    402*   403    406
                         411    413    423*   424    427    432    434    444*   445    448    453    455    465*
                         466    469    474    476    486*   487    490    495    497    507*   508    511    516
                         518    528*   529    532    537    539    549*   550    553    558    560    571*   574
                         583*   585    593*   596    602*   606    617*   619    628*   633    640*   646    653*
                         656    662*   668    675*   677    686*   687*   693    709*   711    761*   762    764
                         772*   777*   778*   779    781    794*   797    803*   806    816*   817    820    825
                         827    837*   838    841    846    848    858*   859    862    867    869    879*   883
                         891*   901*   934*   936    949*   951    963*   965    972*   977    987*   988    990
                        1007*  1008*  1009   1037*  1039   1052*  1056   1073*  1078   1090
ADDR    000006R          164#   252
ADDR22= 001000           211#
ASB     000106R          202#
ASTAT   000104R          200#   307*   308    324*   325    340*   341    349*   350    362*   363*   364    370*
                         371    383*   384*   385    391*   392    404*   405*   406    412*   413    425*   426*
                         427    433*   434    446*   447*   448    454*   455    467*   468*   469    475*   476
                         488*   489*   490    496*   497    509*   510*   511    517*   518    530*   531*   532
                         538*   539    551*   552*   553    559*   560    572*   573*   574    584*   585    595*
                         596    605*   606    618*   619    632*   633    645*   646    655*   656    663*   668
                         676*   677    692*   693    710*   711    763*   764    780*   781    796*   797    805*
                         806    818*   819*   820    826*   827    839*   840*   841    847*   848    860*   861*
                         862    868*   869    882*   883    893*   903*   935*   936    950*   951    964*   965
                         976*   977    989*   990   1006*  1009   1038*  1039   1055*  1056   1077*  1078   1089*
                        1090
AWAS    000110R          203#
BEGIN   000000R          161#   295    311    314    315    328    331    332    344    353    356    357    367
                         374    377    378    388    395    398    399    409    416    419    420    430    437
                         440    441    451    458    461    462    472    479    482    483    493    500    503
                         504    514    521    524    525    535    542    545    546    556    563    566    567
                         577    588    599    609    622    625    626    636    649    659    666    671    680
                         696    699    700    714    726    730    745    748    767    770    771    784    787
                         788    800    809    812    813    823    830    833    834    844    851    854    855
                         865    872    875    876    886    896    906    917    921    939    941    954    968
                         980    993    996    997   1012   1015   1016   1042   1059   1063   1064   1081   1093
                        1097   1098
BITDAT= 010000           241#   654    661    674    685
BITEXT= 004000           242#   594    603    616    630    631    642    643    706    708    721    740    741
                         974
BIT0  = 000001           211#   240    245    305    320    684    761    776    881    890    900    912    986
                        1004   1026
BIT1  = 000002           211#   360    628    629    640    641    691    705    720    739    946    949
BIT10 = 002000           211#
BIT11 = 004000           211#   242
BIT12 = 010000           211#   241    528    705   1071
BIT13 = 020000           211#
BIT14 = 040000           211#   549    739
BIT15 = 100000           211#   628    631    643    740    741   1037   1051   1073   1074   1086
BIT2  = 000004           211#   381    571    615    617    960    963   1030   1037   1052   1072   1073
BIT3  = 000010           211#   402    858
BIT4  = 000020           211#   423    571    661    685    705    837   1003
BIT5  = 000040           211#   444    661    685    690    705    707   1003
BIT6  = 000100           211#   465    720    816    913
BIT7  = 000200           211#   363    384    405    426    447    468    489    510    531    552    573    593
                         594    628    630    631    640    642    643    706    708    721    740    741    819
                         840    861    879    934    949    972    974   1034   1037   1051   1073   1074   1086
```

```
BIT8  = 000400          211#    293     486    1072    1073
BIT9  = 001000          211#    507
BREAK$= 104407          211#    314     315     331     332     356     357     377     378     398     399     419     420
                        440     441     461     462     482     483     503     504     524     525     545     546     566
                        567     625     626     699     700     770     771     787     788     812     813     833     834
                        854     855     875     876     996     997    1015    1016    1063    1064    1097    1098
BR1     000012R         166#    719     738     910
BR2     000013R         167#
BTOD$ = 104421          211#
CDATA$= 104412          211#
CONFIG  000056R         186#
CONT1   000410R         269#    294
CONT2   000420R         272#    754    1102
C$RA    000100R         196#    299*    756*
DATCK$= 104411          211#
DATER$= 104404          211#
DIDIN$  000262R         234#    719*    732     733*
DIDINV  000260R         233#    718*    732*
DIEIN$  000266R         236#    265     738*    750     751*
DIEINV  000264R         235#    737*    750*
DIR     000244R         225#    301     614*    655     663     676     688     689*    692     710     929*    944*    958*
                        983*    989    1001*   1006    1031*   1050*   1070*   1085*
DIR1    000246R         226#
DI1     000542R         303#
DI10    001470R         442#
DI11    001600R         463#
DI12    001710R         484#
DI13    002020R         505#
DI14    002130R         526#
DI15    002240R         547#
DI16    002350R         569#
DI17    002422R         575     581#
DI2     000620R         318#
DI20    002466R         586     592#
DI21    002532R         597     602#
DI22    002574R         607     613#
DI23    002664R         628#
DI24    002740R         634     640#
DI25    003020R         647     653#
DI26    003164R         678     684#
DI27    003302R         705#
DI3     000710R         335#
DI30    003364R         712     718#
DI31    003454R         737#
DI32    004666R         929#
DI33    004756R         937     944#
DI34    005042R         952     958#
DI35    005126R         966     971#
DI36    005202R         978     983#
DI37    005304R        1001#
DI4     001030R         358#
DI40    005414R        1025#
DI41    005636R        1069#
DI5     001140R         379#
DI6     001250R         400#
DI7     001360R         421#
```



```
DODIN$  000256R         231#    260     910*    923     924*
DODINV  000254R         230#    255     909*    923*
DOR     000234R         219#    758     762*    763     779*    780     793*    796     802*    804*    805     880*    889*
                        899*    902*    911*    933*    948*    962*    971*    988*   1005*   1025*   1028*   1035*   1049*
                       1054*   1069*   1076*   1087*
DOR1    000236R         220#    795*
DO0     003564R         753     756#
DO1     003602R         759#
DO10    004470R         884     889#
DO11    004530R         894     899#
DO12    004570R         904     909#
DO2     003660R         774#
DO3     003750R         791#
DO4     004070R         814#
DO5     004200R         835#
DO6     004310R         856#
DO7     004420R         878#
DVID1   000014R         168#    269
ENDIT$= 104413          211#    295
END$  = 104410          211#
ERRTYP  000106R         201#
EXIT$ = 104400          211#    726     745     917
GETPA$= 104415          211#
GWBUF$= 104414          211#
HRDCNT  000044R         181#
HRDER$= 104405          211#    311     328     344     353     367     374     388     395     409     416     430     437
                        451     458     472     479     493     500     514     521     535     542     556     563     577
                        588     599     609     622     636     649     659     666     671     680     696     714     767
                        784     800     809     823     830     844     851     865     872     886     896     906     939
                        954     968     980     993    1012    1042    1059    1081    1093
HRDPAS  000050R         183#
ICONT   000036R         178#
ICOUNT  000040R         179#
ICSR    000240R         223#    250     299     300     361*    362     369*    370     382*    383     390*    391     403*
                        404     411*    412     424*    425     432*    433     445*    446     453*    454     466*    467
                        474*    475     487*    488     495*    496     508*    509     516*    517     529*    530     537*
                        538     550*    551     558*    559     569*    572     581*    582*    584     592*    594*    595
                        603*    604*    605     615*    616*    618     629*    630*    631*    632     641*    642*    643*
                        645     654*    661*    674*    685*    690*    691*    705*    706*    707*    708*    720*    721*
                        728*    739*    740*    741*    746*    931*    935     946*    950     960*    964     974*    975*
                        985*   1003*   1030*   1034*   1038    1051*   1055    1072*   1074*   1077    1086*   1089
ICSR1   000242R         224#    570*    644*
IDNUM   000122R         208#
INIT    000030R         175#
INPUT   000520R         270     299#
INTR    000120R         207#
MAP22$= 104416          211#
MODNAM  000000R         162#
MODSP   000224R         176     209#
MSGN$ = 104403          211#    941
MSGS$ = 104402          211#
MSG$  = 104401          211#
MSG1    006030R         941    1104#
NULL  = 000000          211#    311     328     344     353     367     374     388     395     409     416     430     437
                        451     458     472     479     493     500     514     521     535     542     556     563     577
                        588     599     609     622     636     649     659     666     671     680     696     714     767
```

```
                        784     800     809     823     830     844     851     865     872     886     896     906     939
                        954     968     980     993    1012    1042    1059    1081    1093
OCSR    000230R         216#    246     272     756     757     817*    818     825*    826     838*    839     846*    847
                        859*    860     867*    868     878*    882     892*    893     903     913*    919*    925*    932*
                        947*    961*    973*    976     984*   1002*
OCSR1   000232R         217#    881*    890*    900*    912*
OPEN  = 000000          163     169     170     171     172     189     190     191     192     193     194     195     196
                        198     200     202     203     205     206     207     211#
OTOA$ = 104420          211#
PASCNT  000034R         177#
PIRQ$ = 000004          211#    730     748     921
POPSP = 005726          211#
POPSP2= 022626          211#
PRI1    006034R        1104    1106#
PRTY  = 000000          211#
PRTY0 = 000000          167     211#
PRTY1 = 000040          211#
PRTY2 = 000100          211#
PRTY3 = 000140          211#
PRTY4 = 000200          166     211#
PRTY5 = 000240          211#
PRTY6 = 000300          211#
PRTY7 = 000340          211#
PS    = 177776          211#
PSW   = 177776          211#
PUSH  = 005746          211#
PUSH2 = 024646          211#
RAND$ = 104417          211#
RANNUM  000054R         185#
RESTRT  000306R         204     245#    297
RES1    000056R         187#
RES2    000060R         188#
RSTRT   000112R         204#
SBADR   000102R         197#
SBR     000250R         227#    302     306*    307     323*    324     337*    340     346*    348*    349     613*    930*
                        945*    959*   1029*   1033*   1075*
SBR1    000252R         228#    339*
SOFCNT  000042R         180#
SOFER$= 104406          211#
SOFPA$  000046R         182#
SPOINT  000032R         176#
SPSIZ = 000040            1#    209
SRMINE  000270R         237#    243*    752
SR1     000016R         169#    243
SR2     000020R         170#
SR3     000022R         171#
SR4     000024R         172#
START   000300R         175     243#
STAT    000026R         174#
SVR0    000062R         189#
SVR1    000064R         190#
SVR2    000066R         191#
SVR3    000070R         192#
SVR4    000072R         193#
SVR5    000074R         194#
SVR6    000076R         195#
```



```
SYSCNT  000052R         184#
TEMP    000272R         238#    320*    321     333*    684*    686     689     701*    776*    777     789*    986*    987
                        998*   1004*   1005    1007    1017*   1026*   1033    1035    1047    1065*   1071*   1075    1076
                       1087    1099#
TEMP1   000274R         239#   1047*   1048*   1054
TEMP2   000276R         240#    245*    269     292*    293     752
TRPDFD= 000022          211#
VECTOR  000010R         165#    262
WASADR  000104R         199#
WDFR    000116R         206#
WDTO    000114R         205#
XFLAG   000005R         163#
$BASE1  000224R         212#    247
$VECT1  000226R         213#    257


. ABS.  000000     000
        006112     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XMNBA0,XMNBA0/SOL/CRF:SYM=DDXCOM,XMNBA0
 RUN-TIME:  2  4  .5 SECONDS
 RUN-TIME RATIO:  185/8=21.5
 CORE USED:  7K  (13 PAGES)
```