

.REM

IDENTIFICATION

PRODUCT CODE: AC-F419A-MC
PRODUCT NAME: CXMNB00 MNCDI MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1979, 1979 DIGITAL EQUIPMENT CORPORATION

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 VERIFYING 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 SRI = 0 IF SRI = 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 (SRI=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

SRI = 0 RUN MNCDI LOGIC TEST
SRI = 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 THAN A 1, MULTIPLE MNCDI
WILL BE TESTED. IF DEVCNT (DVID1) CONTAINS MORE THAN 1 AND SRI=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.

```

155 .LIST MF
156 .NLIST MC,CND,MD
157 .TITLE MNBA DEC/X11 SYSTEM EXERCISER MODULE
158 ; DDXCOM VERSION 6 23-MAY-78
159 .LIST FIN
160 ;*****
161 000000* REGIN:
162 000000* 047115 040502 040 MODNAM: .ASCII /MNBA /MODULE NAME,
163 000005* 000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUF USAGE
164 000006* 171160 ADDR: 171160+0 ;1ST DEVICE ADDR.
165 000010* 000130 VECTOR: 130+0 ;1ST DEVICE VECTOR,
166 000012* 200 RR1: .BYTE PRTY4+0 ;1ST RR LEVEL.
167 000013* 000 RR2: .BYTE PRTY0+0 ;2ND RR LEVEL.
168 000014* 000001 DVID1: 0+1 ;DEVICE INDICATOR 1.
169 000016* 000000 SR1: OPEN ;SWITCH REGISTER 1
170 000020* 000000 SR2: OPEN ;SWITCH REGISTER 2
171 000022* 000000 SR3: OPEN ;SWITCH REGISTER 3
172 000024* 000000 SR4: OPFN ;SWITCH REGISTER 4
173 ;*****
174 000026* 140000 STAT: 140000 ;STATUS WORD.
175 000030* 000300 INIT: START ;MODULE START ADDR.
176 000032* 000224* SPOINT: MODSP ;MODULE STACK POINTER,
177 000034* 000000 PASCNT: 0 ;PASS COUNTER.
178 000036* 002000 ICNT: 2000 ;# OF ITERATIONS PER PASS=2000
179 000040* 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
180 000042* 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
181 000044* 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
182 000046* 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
183 000050* 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
184 000052* 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
185 000054* 000000 PANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
186 000056* RES1: 0 ;RESERVED FOR MONITOR USE
187 000056* 000000 RES2: 0 ;RESERVED FOR MONITOR USE
188 000060* 000000 SVR0: OPEN ;LOC TO SAVE R0.
189 000062* 000000 SVR1: OPEN ;LOC TO SAVE R1.
190 000064* 000000 SVR2: OPEN ;LOC TO SAVE R2.
191 000066* 000000 SVR3: OPEN ;LOC TO SAVE R3.
192 000070* 000000 SVR4: OPFN ;LOC TO SAVE R4.
193 000072* 000000 SVR5: OPEN ;LOC TO SAVE R5.
194 000074* 000000 SVR6: OPEN ;LOC TO SAVE R6.
195 000076* 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
196 00102* SBADR: ;ADDR OF GOOD DATA, OR
197 00102* 000000 ACSR: OPEN ;CONTENTS OF CSR.
198 00104* WABADR: ;ADDR OF BAD DATA, OR
199 00104* 000000 ASTAT: OPEN ;STATUS REG CONTENTS.
200 00106* FRRTYP: ;TYPE OF EPROP.
201 00106* 000000 ASB: OPEN ;EXPECTED DATA.
202 00110* 000000 AWAS: OPEN ;ACTUAL DATA.
203 00112* 000306* RSTPT: RESTPT ;RESTART ADDRESS AFTER END OF PASS
204 00114* 000000 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
205 00116* 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
206 00120* 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
207 00122* 000000 IDNUM: 0 ;MODULE IDENTIFICATION NUMBER=0

```

```

209 000224* MODSP:
210 ;*****
211
212 000224* 171260 $BASE1: 171260 ;INITIAL BUS ADDRESS OF THE MNCDO <IF SR1=1>
213 000226* 000340 SVFCT1: 340 ;INITIAL INTERRUPT VECTOR OF THE MNCDO <IF SR1=1>
214
215 ;OUTPUT ADDRESS
216 000230* 000000 OCSR: 0
217 000232* 000000 OCSR1: 0 ;HIGH BYTE ADDRESS
218
219 000234* 000000 DOR: 0
220 000236* 000000 DOPI: 0 ;HIGH BYTE ADDRESS
221
222 ;INPUT ADDRESS
223 000240* 000000 ICSR: 0
224 000242* 000000 ICSR1: 0 ;HIGH BYTE ADDRESS
225 000244* 000000 DIR: 0
226 000246* 000000 DIR1: 0
227 000250* 000000 SBR: 0
228 000252* 000000 SBR1: 0 ;HIGH BYTE ADDRESS
229
230 000254* 000000 DODINV: 0
231 000256* 000000 DODINS: 0
232
233 000260* 000000 DIDINV: 0
234 000262* 000000 DIDINS: 0
235 000264* 000000 DIEINV: 0
236 000266* 000000 DIEINS: 0
237 000270* 000000 SRMINE: 0 ;MY COPY OF SR1
238 000272* 000000 TEMP: 0
239 000274* 000000 TEMP1: 0
240 000276* 000001 TEMP2: BIT0
241 010000 BITDAT=BIT12 ;MAINT INPUT INHIBIT
242 004000 BITEXT=BIT11 ;MAINT INPUT STROBE

```

```

243 000300* 016767 177512 177762 START: MOV S#1,SPMINE ;COPY SR1 FOR MY USE
244 ;INITILIZE THE BUS ADDRESSES AND VECTORS
245 000306* 012767 000000* 177762 RESTR1: MOV #R10,TEMP2 ;LOAD UNIT SELECT FLAG
246 000314* 012700 000230* MOV #ACSR,R0 ;LOAD ADDRESS POINTER
247 000320* 016701 177700 MOV $ASE1,R1 ;LOAD INITIAL BUS ADDRESS
248 000324* 010120 16: MOV R1,(R0)+ ;LOAD DEVICE ADDRESS
249 000326* 005201 INC R1 ;UPDATE BUS ADDRESS VALUE
250 000330* 020027 000240* CMP R0,#ICSR ;TEST IF DONE WITH BUS ADDRESSES
251 000334* 001373 BNE 16 ;JRR IF NOT
252 000336* 016701 177444 MOV ADDR,R1 ;LOAD INPUT ADDRESS
253 000342* 010120 26: MOV R1,(R0)+ ;LOAD THE ADDRESS
254 000344* 005201 INC P1 ;UPDATE ADDRESS
255 000346* 020027 000254* CMP R0,#DODINV ;TEST IF AT END
256 000352* 001373 BNE 26 ;JRR IF NOT
257 000354* 016701 177646 MOV $VECT1,R1 ;LOAD VECTOR POINTER
258 000360* 010120 36: MOV R1,(R0)+ ;LOAD DEVICE VECTOR ADDRESS
259 000362* 005721 TST (R1)+ ;UPDATE BUS VECTOR VALUE
260 000364* 020027 000260* CMP R0,#DODINS+2 ;TEST IF DONE WITH BUS VECTORS
261 000370* 001373 BNE 36 ;JRR IF NOT
262 000372* 016701 177412 MOV $VECTOR,R1 ;LOAD INPUT VECTOR
263 000376* 010120 46: MOV R1,(R0)+ ;LOAD THE VECTOR
264 000400* 005721 TST (R1)+ ;BUMP THE ADDRESS
265 000402* 020027 000270* CMP R0,#DIEINS+2 ;TEST IF DONE
266 000406* 001373 BNE 46 ;JRR IF NOT
267
268
269 000410* 036767 177662 177376 ;TFST IF UNIT IS TO RUN
270 000416* 001040 CONT1: BIT TEMP2,DVID1 ;IS UNIT SELECTED ?
271 ;UNIT IS NOT SLEPCTED == CORPECT THE ADDRESSES
272 000420* 012700 000230* CONT2: MOV #ACSR,R0 ;GET ADDRESS
273 000424* 012701 000004 MOV #4,R1 ;GET NEW OFFSET VALUE
274 000430* 060110 ADD R1,(R0) ;UPDATE THE VALUE
275 000432* 060110 ADD R1,(R0) ;UPDATE THE VALUE
276 000434* 060110 ADD R1,(R0) ;UPDATE THE VALUE
277 000436* 060110 ADD R1,(R0) ;UPDATE THE VALUE
278 000440* 012701 000010 MOV #0,R1 ;RELOAD NEW OFFSET
279 000444* 060110 ADD R1,(R0) ;UPDATE THE VALUE
280 000446* 060110 ADD R1,(R0) ;UPDATE THE VALUE
281 000450* 060110 ADD R1,(R0) ;UPDATE THE VALUE
282 000452* 060110 ADD R1,(R0) ;UPDATE THE VALUE
283 000454* 060110 ADD R1,(R0) ;UPDATE THE VALUE
284 000456* 060110 ADD R1,(R0) ;UPDATE THE VALUE
285 000460* 060110 ADD R1,(R0) ;UPDATE THE VALUE
286 000462* 060110 ADD R1,(R0) ;UPDATE THE VALUE
287 000464* 060110 ADD R1,(R0) ;UPDATE THE VALUE
288 000466* 060110 ADD R1,(R0) ;UPDATE THE VALUE
289 000470* 060110 ADD R1,(R0) ;UPDATE THE VALUE
290 000472* 060110 ADD R1,(R0) ;UPDATE THE VALUE
291
292 000474* 006367 177576 ;DETERMINE IF NEXT UNIT IS TO BE TESTED
293 000500* 022767 000400 177570 ASL TEMP2 ;CHANGE UNITS
294 000506* 001340 CMP #R10,TEMP2 ;TEST IF LAST UNIT
295 000510* 104413 000000* BNE CONT1 ;JRR IF NOT
296
297 000514* 000167 177566 ENDITS,REGIN ;SIGNAL END OF ITERATION.
;MONITOR SHALL TEST END OF PASS
JMP RESTR1

```

```

298 ;VERIFY A MNCDI BUS ADDRESS RESPONSE
299 000520* 016767 177514 177352 INPUT: MOV ICSR,CSPA ;LOAD ADDR.
300 000526* 005777 177506 TST #ICSR ;TEST INPUT STATUS
301 000532* 005777 177506 TST #DIR ;TEST INPUT DATA REGISTER
302 000536* 005777 177506 TST #SBR ;TEST STIM. BUFFER REGISTER
303 000542*
304
305 000542* 012767 000001 177332 DI1: ;FLOAT A 1 ACROSS THE MNCDI STIMULUS BIT REGISTER
306 000550* 016777 177326 177472 MOV #R10,ACSR ;LOAD EXPECT BIT
307 000556* 017767 177466 177320 16: MOV ACSR,#SBR ;LOAD MNCDI STIMULUS BIT REGISTER
308 000564* 026767 177312 177312 MOV #SBR,ASTAT ;READ MNCDI STIMULUS BIT REGISTER
309 000572* 001403 CMP ACSR,ASTAT ;COMPARE
310 REQ 26 ;JRR IF SAME
311 000574* 104405 000000* 000000 *****
312 HDRDR#,REGIN,NULL ;MNCDI STIMULUS BIT REGISTER FAILED TO HOLD A FLOATING
313 *****
314 000602* 104407 000000* 26: BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....
315 000606* 104407 000000* BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
316 000612* 006367 177264 ASL ACSR ;CHANGE THE DATA
317 000616* 001354 BNE 16 ;JRR IF MORE DATA
318 000620*
319
320 000620* 012767 000001 177444 DI2: ;FLOAT A 0 ACROSS THE MNCDI STIMULUS BIT REGISTER
321 000626* 016767 177440 177246 16: MOV #R10,TEMP ;LOAD INITIAL BIT
322 000634* 005167 177242 MOV TEMP,ACSP ;LOAD EXPECTED
323 000640* 016777 177236 177402 COM ACSR ;COMPLEMENT
324 000646* 017767 177376 177230 MOV ACSR,#SBR ;LOAD MNCDI STIMULUS BIT REGISTER
325 000654* 026767 177222 177222 MOV #SBR,ASTAT ;READ MNCDI STIMULUS BIT REGISTER
326 000662* 001403 CMP ACSR,ASTAT ;COMPARE
327 REQ 26 ;JRR IF SAME
328 000664* 104405 000000* 000000 *****
329 HDRDR#,REGIN,NULL ;MNCDI STIMULUS BIT REGISTER FAILED TO HOLD A FLOATING 0
330 *****
331 000672* 104407 000000* 26: BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....
332 000676* 104407 000000* BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
333 000702* 006367 177364 ASL TEMP ;CHANGE THE DATA
334 000706* 001347 BNE 16 ;JRR IF MORE DATA
335 000710*
336
337 000710* 012777 177777 177332 DI3: ;VERIFY BYTE OPERATION ON THE MNCDI STIMULUS BIT REGISTER
338 000716* 012767 000377 177156 MOV #-1,#SBR ;LOAD MNCDI STIMULUS BIT REGISTER
339 000724* 105077 177322 CLR# #SBR1 ;LOAD EXPECTED
340 000730* 017767 177314 177146 MOV #SBR,ASTAT ;CLEAR HIGH BYTE
341 000736* 026767 177140 177140 CMP ACSR,ASTAT ;READ MNCDI STIMULUS BIT REGISTER
342 000744* 001403 REQ 26 ;COMPARE
343 ;JRR IF SAME
344 000746* 104405 000000* 000000 *****
345 HDRDR#,REGIN,NULL ;CLEARING HIGH BYTE CHANGED LOW BYTE
346 *****
346 000754* 012777 177777 177266 26: MOV #-1,#SBR ;LOAD MNCDI STIMULUS BIT REGISTER
347 000762* 012767 177400 177112 MOV #177400,ACSR ;LOAD EXPECTED
348 000770* 105077 177254 CLR# #SBR ;CLEAR LOW BYTE
349 000774* 017767 177250 MOV #SBR,ASTAT ;READ MNCDI STIMULUS BIT REGISTER
350 001002* 026767 177074 177074 CMP ACSR,ASTAT ;COMPARE
351 001010* 001403 REQ 36 ;JRR IF SAME
352 *****
353 001012* 104405 000000* 000000 HDRDR#,REGIN,NULL ;CLEARING LOW BYTE CHANGED HIGH BYTE

```

```
354  
355 001020  
356 001020 104407 000000  
357 001024 104407 000000  
358 001030  
359  
360 001030 012767 000002 177044  
361 001036 016777 177040 177174  
362 001044 017767 177170 177032  
363 001052 042767 000200 177024  
364 001060 026767 177016 177016  
365 001066 001403  
366  
367 001070 104405 000000 000000  
368  
369 001076 046777 177000 177134  
370 001104 017767 177130 176772  
371 001112 026767 176764 176764  
372 001120 001003  
373  
374 001122 104405 000000 000000  
375  
376 001130  
377 001130 104407 000000  
378 001134 104407 000000  
379 001140  
380  
381 001140 012767 000004 176734  
382 001146 016777 176730 177064  
383 001154 017767 177060 176722  
384 001162 042767 000200 176714  
385 001170 026767 176706 176706  
386 001176 001403  
387  
388 001200 104405 000000 000000  
389  
390 001206 046777 176670 177024  
391 001214 017767 177020 176662  
392 001222 026767 176654 176654  
393 001230 001003  
394  
395 001232 104405 000000 000000  
396  
397 001240  
398 001240 104407 000000  
399 001244 104407 000000
```

361
BREAK\$,BEGIN ;TEMPORARY RETURN TO MONITOR...
BREAK\$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.

DI4:
;TEST THAT BIT1 OF MNCDI STATUS REGISTER IS READ-WRITE
MOV #BIT1,ACSR ;LOAD EXPECTED
MOV ACSR,@ICSR ;LOAD BIT1 INTO MNCDI STATUS REGISTER
MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER
BIC #BIT7,ASTAT ;CLEAR BIT 7
CMP ACSR,ASTAT ;TEST THAT IT SET
BEQ 1\$;BR IF SAME
;*****
HRDR\$,BEGIN,NULL ;BIT1 OF MNCDI STATUS REGISTER FAILED TO SET
;*****
1\$:
BIC ACSR,@ICSR ;CLEAR THAT BIT
MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
CMP ACSR,ASTAT ;TEST THE BIT
BNE 2\$;BR IF CLEARED
;*****
2\$:
BREAK\$,BEGIN ;TEMPORARY RETURN TO MONITOR...
BREAK\$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.

DI5:
;TEST THAT BIT2 OF MNCDI STATUS REGISTER IS READ-WRITE
MOV #BIT2,ACSR ;LOAD EXPECTED
MOV ACSR,@ICSR ;LOAD BIT2 INTO MNCDI STATUS REGISTER
MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER
BIC #BIT7,ASTAT ;CLEAR BIT 7
CMP ACSR,ASTAT ;TEST THAT IT SET
BEQ 1\$;BR IF SAME
;*****
HRDR\$,BEGIN,NULL ;BIT2 OF MNCDI STATUS REGISTER FAILED TO SET
;*****
1\$:
BIC ACSR,@ICSR ;CLEAR THAT BIT
MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
CMP ACSR,ASTAT ;TEST THE BIT
BNE 2\$;BR IF CLEARED
;*****
2\$:
BREAK\$,BEGIN ;TEMPORARY RETURN TO MONITOR...
BREAK\$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.

```
400 001250  
401  
402 001250 012767 000010 176624  
403 001256 016777 176620 176754  
404 001264 017767 176750 176612  
405 001272 042767 000200 176604  
406 001300 026767 176576 176576  
407 001306 001403  
408  
409 001310 104405 000000 000000  
410  
411 001316 046777 176560 176714  
412 001324 017767 176710 176552  
413 001332 026767 176544 176544  
414 001340 001003  
415  
416 001342 104405 000000 000000  
417  
418 001350  
419 001350 104407 000000  
420 001354 104407 000000  
421 001360  
422  
423 001360 012767 000020 176514  
424 001366 016777 176510 176644  
425 001374 017767 176640 176502  
426 001402 042767 000200 176474  
427 001410 026767 176466 176466  
428 001416 001403  
429  
430 001420 104405 000000 000000  
431  
432 001426 046777 176450 176604  
433 001434 017767 176600 176442  
434 001442 026767 176434 176434  
435 001450 001003  
436  
437 001452 104405 000000 000000  
438  
439 001460  
440 001460 104407 000000  
441 001464 104407 000000
```

DI6:
;TEST THAT BIT3 OF MNCDI STATUS REGISTER IS READ-WRITE
MOV #BIT3,ACSR ;LOAD EXPECTED
MOV ACSR,@ICSR ;LOAD BIT3 INTO MNCDI STATUS REGISTER
MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER
BIC #BIT7,ASTAT ;CLEAR BIT 7
CMP ACSR,ASTAT ;TEST THAT IT SET
BEQ 1\$;BR IF SAME
;*****
HRDR\$,BEGIN,NULL ;BIT3 OF MNCDI STATUS REGISTER FAILED TO SET
;*****
1\$:
BIC ACSR,@ICSR ;CLEAR THAT BIT
MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
CMP ACSR,ASTAT ;TEST THE BIT
BNE 2\$;BR IF CLEARED
;*****
2\$:
BREAK\$,BEGIN ;TEMPORARY RETURN TO MONITOR...
BREAK\$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.

DI7:
;TEST THAT BIT4 OF MNCDI STATUS REGISTER IS READ-WRITE
MOV #BIT4,ACSR ;LOAD EXPECTED
MOV ACSR,@ICSR ;LOAD BIT4 INTO MNCDI STATUS REGISTER
MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER
BIC #BIT7,ASTAT ;CLEAR BIT 7
CMP ACSR,ASTAT ;TEST THAT IT SET
BEQ 1\$;BR IF SAME
;*****
HRDR\$,BEGIN,NULL ;BIT4 OF MNCDI STATUS REGISTER FAILED TO SET
;*****
1\$:
BIC ACSR,@ICSR ;CLEAR THAT BIT
MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
CMP ACSR,ASTAT ;TEST THE BIT
BNE 2\$;BR IF CLEARED
;*****
2\$:
BREAK\$,BEGIN ;TEMPORARY RETURN TO MONITOR...
BREAK\$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.

```
442 001470* DI10:
443
444 001470* 012767 000040 176404 ;TEST THAT BITS OF MNCDI STATUS REGISTER IS READ-WRITE
445 001476* 016777 176400 176534 MOV #BITS,ACSR ;LOAD EXPECTED
446 001504* 017767 176530 176372 MOV ACSR,ICSR ;LOAD BITS INTO MNCDI STATUS REGISTER
447 001512* 042767 000200 176364 MOV ICSP,ASTAT ;READ MNCDI STATUS REGISTER
448 001520* 026767 176356 176356 BIC #BIT7,ASTAT ;CLEAR BIT 7
449 001526* 001403 CMP ACSR,ASTAT ;TEST THAT IT SET
450 BEQ 18 ;BR IF SAME
451 001530* 104405 000000* 000000 ;*****
452 HRDR$,BEGIN,NULL ;BITS OF MNCDI STATUS REGISTER FAILED TO SET
453 001536* 046777 176340 176474 18: BIC ACSR,ICSR ;CLEAR THAT BIT
454 001544* 017767 176470 176332 MOV ICSP,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
455 001552* 026767 176324 176324 CMP ACSR,ASTAT ;TEST THE BIT
456 001560* 001003 BNE 28 ;BR IF CLEARED
457
458 001562* 104405 000000* 000000 ;*****
459 HRDR$,BEGIN,NULL ;BITS OF MNCDI STATUS REGISTER FAILED TO CLEAR
460 001570* 28: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
461 001574* 104407 000000* BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
462
463 001600* DI11:
464
465 001600* 012767 000100 176274 ;TEST THAT BIT6 OF MNCDI STATUS REGISTER IS READ-WRITE
466 001606* 016777 176270 176424 MOV #BIT6,ACSR ;LOAD EXPECTED
467 001614* 017767 176420 176252 MOV ACSR,ICSR ;LOAD BITS INTO MNCDI STATUS REGISTER
468 001622* 042767 000200 176254 MOV ICSP,ASTAT ;READ MNCDI STATUS REGISTER
469 001630* 026767 176246 176246 BIC #BIT7,ASTAT ;CLEAR BIT 7
470 001636* 001403 CMP ACSR,ASTAT ;TEST THAT IT SET
471 BEQ 18 ;BR IF SAME
472 001640* 104405 000000* 000000 ;*****
473 HRDR$,BEGIN,NULL ;BIT6 OF MNCDI STATUS REGISTER FAILED TO SET
474 001646* 046777 176230 176364 18: BIC ACSR,ICSR ;CLEAR THAT BIT
475 001654* 017767 176360 176222 MOV ICSP,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
476 001662* 026767 176214 176214 CMP ACSR,ASTAT ;TEST THE BIT
477 001670* 001003 BNE 28 ;BR IF CLEARED
478
479 001672* 104405 000000* 000000 ;*****
480 HRDR$,BEGIN,NULL ;BIT6 OF MNCDI STATUS REGISTER FAILED TO CLEAR
481 001700* 28: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
482 001704* 104407 000000* BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
483 001704* 104407 000000*
```

```
484 001710* DI12:
485
486 001710* 012767 000400 176164 ;TEST THAT BITS OF MNCDI STATUS REGISTER IS READ-WRITE
487 001716* 016777 176160 176314 MOV #BIT8,ACSR ;LOAD EXPECTED
488 001724* 017767 176310 176152 MOV ACSR,ICSR ;LOAD BITS INTO MNCDI STATUS REGISTER
489 001732* 042767 000200 176144 MOV ICSP,ASTAT ;READ MNCDI STATUS REGISTER
490 001740* 026767 176136 176136 BIC #BIT7,ASTAT ;CLEAR BIT 7
491 001746* 001403 CMP ACSR,ASTAT ;TEST THAT IT SET
492 BEQ 18 ;BR IF SAME
493 001750* 104405 000000* 000000 ;*****
494 HRDR$,BEGIN,NULL ;BITS OF MNCDI STATUS REGISTER FAILED TO SET
495 001756* 046777 176120 176254 18: BIC ACSR,ICSR ;CLEAR THAT BIT
496 001764* 017767 176250 176112 MOV ICSP,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
497 001772* 026767 176104 176104 CMP ACSR,ASTAT ;TEST THE BIT
498 002000* 001003 BNE 28 ;BR IF CLEARED
499
500 002002* 104405 000000* 000000 ;*****
501 HRDR$,BEGIN,NULL ;BITS OF MNCDI STATUS REGISTER FAILED TO CLEAR
502 002010* 28: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
503 002014* 104407 000000* BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
504 002014* 104407 000000*
505 002020* DI13:
506
507 002020* 012767 001000 176054 ;TEST THAT BIT9 OF MNCDI STATUS REGISTER IS READ-WRITE
508 002026* 016777 176050 176204 MOV #BIT9,ACSR ;LOAD EXPECTED
509 002034* 017767 176200 176042 MOV ACSR,ICSR ;LOAD BITS INTO MNCDI STATUS REGISTER
510 002042* 042767 000200 176034 MOV ICSP,ASTAT ;READ MNCDI STATUS REGISTER
511 002050* 026767 176026 176026 BIC #BIT7,ASTAT ;CLEAR BIT 7
512 002056* 001403 CMP ACSR,ASTAT ;TEST THAT IT SET
513 BEQ 18 ;BR IF SAME
514 002060* 104405 000000* 000000 ;*****
515 HRDR$,BEGIN,NULL ;BIT9 OF MNCDI STATUS REGISTER FAILED TO SET
516 002066* 046777 176010 176144 18: BIC ACSR,ICSR ;CLEAR THAT BIT
517 002074* 017767 176140 176002 MOV ICSP,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
518 002102* 026767 175774 175774 CMP ACSR,ASTAT ;TEST THE BIT
519 002110* 001003 BNE 28 ;BR IF CLEARED
520
521 002112* 104405 000000* 000000 ;*****
522 HRDR$,BEGIN,NULL ;BIT9 OF MNCDI STATUS REGISTER FAILED TO CLEAR
523 002120* 28: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
524 002124* 104407 000000* BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
525 002124* 104407 000000*
```

```
526 002130  DI14: ;TEST THAT BIT12 OF MNCDI STATUS REGISTER IS READ-WRITE
527 ;
528 002130 012767 010000 175744 MOV #BIT12,ACSR ;LOAD EXPECTED
529 002136 016777 175740 176074 MOV ACSR,@ICSR ;LOAD BIT12 INTO MNCDI STATUS REGISTER
530 002144 017767 176070 175732 MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER
531 002152 042767 000200 175724 BIC #BIT7,ASTAT ;CLEAR BIT 7
532 003160 026767 175716 175716 CMP ACSR,ASTAT ;TEST THAT IT SET
533 002166 001403 BEQ 18 ;BR IF SAME
534 ;
535 002170 104405 000000 000000 ;*****
536 HRDR$,BEGIN,NULL ;BIT12 OF MNCDI STATUS REGISTER FAILED TO SET
537 ;*****
538 002176 046777 175700 176034 18: BIC ACSR,@ICSR ;CLEAR THAT BIT
539 002204 017767 176030 175672 MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
540 002212 026767 175664 175664 CMP ACSR,ASTAT ;TEST THE BIT
541 002220 001003 BNE 28 ;BR IF CLEARED
542 ;
543 002222 104405 000000 000000 ;*****
544 HRDR$,BEGIN,NULL ;BIT12 OF MNCDI STATUS REGISTER FAILED TO CLEAR
545 ;*****
546 002230 104407 000000 28: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR...
547 002234 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
548 ;
549 002240 012767 040000 175634 DI15: ;TEST THAT BIT14 OF MNCDI STATUS REGISTER IS READ-WRITE
550 002246 016777 175630 175764 MOV #BIT14,ACSR ;LOAD EXPECTED
551 002254 017767 175760 175622 MOV ACSR,@ICSR ;LOAD BIT14 INTO MNCDI STATUS REGISTER
552 002262 042767 000200 175614 MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER
553 002270 026767 175606 175606 BIC #BIT7,ASTAT ;CLEAR BIT 7
554 002276 001403 CMP ACSR,ASTAT ;TEST THAT IT SET
555 BEQ 18 ;BR IF SAME
556 ;
557 002300 104405 000000 000000 ;*****
558 HRDR$,BEGIN,NULL ;BIT14 OF MNCDI STATUS REGISTER FAILED TO SET
559 ;*****
560 002306 046777 175570 175724 18: BIC ACSR,@ICSR ;CLEAR THAT BIT
561 002314 017767 175720 175562 MOV @ICSR,ASTAT ;READ MNCDI STATUS REGISTER AGAIN
562 002322 026767 175554 175554 CMP ACSR,ASTAT ;TEST THE BIT
563 002330 001003 BNE 28 ;BR IF CLEARED
564 ;
565 002332 104405 000000 000000 ;*****
566 HRDR$,BEGIN,NULL ;BIT14 OF MNCDI STATUS REGISTER FAILED TO CLEAR
567 ;*****
568 002340 104407 000000 28: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR...
569 002344 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
```

```
568 ;VERIFY HIGH BYTE OPERATION ON THE INPUT STATUS REGISTER
569 002350 012777 040424 175662 DI16: MOV #40424,@ICSR ;LOAD INPUT REG. BIT
570 002356 105077 175660 CLR @ICSR1 ;CLEAR HIGH BYTE
571 002362 012767 000024 175512 MOV #BIT4|BIT2,ACSR ;LOAD EXPECTED
572 002370 017767 175644 175506 MOV @ICSR,ASTAT ;READ INPUT STATUS REG.
573 002376 042767 000200 175500 BIC #BIT7,ASTAT ;REMOVE BIT 7
574 002404 026767 175472 175472 CMP ACSR,ASTAT ;COMPARE
575 002412 001403 BEQ DI17
576 ;
577 002414 104405 000000 000000 ;*****
578 HRDR$,BEGIN,NULL ;CLEARING HIGH BYTE CHANGED LOW BYTE
579 ;*****
580 ;VERIFY LOW BYTE OPERATION ON THE INPUT STATUS REGISTER
581 002422 012777 040426 175610 DI17: MOV #40426,@ICSR ;LOAD INPUT REG.
582 002430 105077 175604 CLR @ICSR ;CLEAR LOW BYTE
583 002434 012767 040400 175440 MOV #40400,ACSR ;LOAD EXPECTED
584 002442 017767 175572 175434 MOV @ICSR,ASTAT ;READ INPUT STATUS REG.
585 002450 026767 175426 175426 CMP ACSR,ASTAT ;COMPARE
586 002456 001403 BEQ DI20 ;BR IF SAME
587 ;
588 002460 104405 000000 000000 ;*****
589 HRDR$,BEGIN,NULL ;CLEARING LOW BYTE CHANGED HIGH BYTE
590 ;*****
591 ;VERIFY THAT MAINT. STROBE SETS "INPUT DATA READY"
592 002466 005077 175546 DI20: CLR @ICSR ;ENSURE CLEAR FLAG
593 002472 012767 000200 175402 MOV #BIT7,ACSR ;LOAD EXPECTED DATA
594 002500 012777 004200 175532 MOV #BIT7|BIT7,@ICSR ;GENERATE MAINT. STROBE
595 002506 017767 175526 175370 MOV @ICSR,ASTAT ;READ INPUT STATUS REGISTER
596 002514 026767 175362 175362 CMP ACSR,ASTAT ;COMPARE RESULTS
597 002522 001403 BEQ DI21 ;BR IF SAME
598 ;
599 002524 104405 000000 000000 ;*****
600 HRDR$,BEGIN,NULL ;MAINT. STROBE FAILED TO SET "INPUT DATA READY"
;*****
```



```

601 ;VERIFY THAT "INPUT DATA READY" CAN BE WRITTEN TO A ZERO
602 CLR ACSR ;LOAD EXPECTED DATA
603 MOV #BITEXT,@ICSR ;GENERATE MAINT, STROBE
604 CLR @ICSR ;CLEAR DATA READY FLAG
605 MOV @ICSR,ASTAT ;READ INPUT STATUS REGISTER
606 CMP ACSR,ASTAT ;COMPARE
607 BEQ DI22 ;BR IF SAME
608 ;*****
609 HRDR#,BEGIN,NULL ;"INPUT DATA READY" FAILED TO BE WRITTEN TO A ZERO
610 ;*****
611
612 ;VERIFY THAT "INPUT DATA READY" WILL NOT SET IF IN STIMULUS MODE AND NO SBR MATC
613 CLR @SBR ;CLEAR SBR REGISTER
614 MOV #-1,@DIR ;CLEAR INPUT REGISTER
615 MOV #BIT2,@ICSR ;SET STILULUS MODE
616 BIS #BITEXT,@ICSR ;GENERATE MAINT, STROBE
617 MOV #BIT2,ACSR ;LOAD EXPECTED
618 MOV @ICSR,ASTAT ;READ STATUS
619 CMP ACSR,ASTAT ;COMPARE
620 BEQ 1$ ;BR IF SAME
621 ;*****
622 HRDR#,BEGIN,NULL ;INPUT STROBE SET INPUT READY WHEN IN STIMULUS MODE
623 ;*****
624
625 181 BREAK#,BEGIN ;TEMPORARY RETURN TO MONITOR,...
626 002660# 104400# 000000# ;THEN CONTINUE AT NEXT INSTRUCTION.
627 ;VERIFY THAT "OVERRUN ERROR" SETS
628 MOV #BIT15|BIT7|BIT1,ACSR ;LOAD EXPECTED
629 MOV #BIT1,@ICSR ;SET STROBE MODE
630 BIS #BITEXT|BIT7,@ICSR ;GENERATE MAINT, STROBE
631 BIS #BIT15|BITEXT|BIT7,@ICSR ;GENERATE MAINT, STROBE AGAIN
632 MOV @ICSR,ASTAT ;READ INPUT STATUS REGISTER
633 CMP ACSR,ASTAT ;COMPARE
634 BEQ DI24 ;BR IF SAME
635 ;*****
636 HRDR#,BEGIN,NULL ;"OVER RUN" FAILED TO SET
637 ;*****
638
639 ;VERIFY THAT "OVERRUN ERROR" CAN BE WRITTEN TO A ZERO
640 MOV #BIT7|BIT1,ACSR ;LOAD EXPECTED VALUE
641 MOV #BIT1,@ICSR ;SET STROBE MODE
642 BIS #BITEXT|BIT7,@ICSR ;GENERATE MAINT, STROBE
643 BIS #BIT15|BITEXT|BIT7,@ICSR ;GENERATE MAINT, STROBE AGAIN
644 CLR @ICSR1 ;CLEAR HIGH BYTE OF THE INPUT STATUS REGISTER
645 MOV @ICSR,ASTAT ;READ INPUT STATUS REGISTER
646 CMP ACSR,ASTAT ;COMPARE
647 BEQ DI25 ;BR IF SAME
648 ;*****
649 HRDR#,BEGIN,NULL ;"OVERRUN ERROR" FAILED TO BE WRITTEN TO A ZERO
650 ;*****
651

```

```

652 ;VERIFY INVERT DATA FUNCTION
653 CLR ACSR ;LOAD EXPECTED
654 MOV #BITDAT,@ICSR ;SET INPUT INHIBIT
655 MOV @DIR,ASTAT ;READ INPUT
656 CMP ACSR,ASTAT ;COMPARE
657 BEQ 1$ ;BR IF SAME
658 ;*****
659 HRDR#,BEGIN,NULL ;INPUT INHIBIT FAILED TO INHIBIT INPUT
660 ;*****
661 MOV #BITDAT|BIT5|BIT4,@ICSR ;SET INVERT DATA AND INPUT INHIBIT
662 MOV #-1,ACSR ;LOAD EXPECTED
663 MOV @DIR,ASTAT ;READ INPUT
664 BNE 2$ ;BR IF NON-ZERO
665 ;*****
666 HRDR#,BEGIN,NULL ;INVERT DATA FUNCTION FAILED
667 ;*****
668 CMP ACSR,ASTAT ;COMPARE DATA
669 BEQ 3$ ;BR IF SAME
670 ;*****
671 HRDR#,BEGIN,NULL ;INVERT DATA - DATA PATH ERROR
672 ;*****
673
674 361 MOV #BITDAT,@ICSR ;SET INPUT INHIBIT
675 CLR ACSR ;CLEAR EXPECTED
676 MOV @DIR,ASTAT ;READ INPUT
677 CMP ACSR,ASTAT ;COMPARE
678 BEQ DI26 ;BR IF SAME
679 ;*****
680 HRDR#,BEGIN,NULL ;INVERT DATA FUNCTION OR INPUT INHIBIT FAILED
681 ;*****
682
683 ;VERIFY EACH BIT OF THE MNCDI INPUT DATA REGISTER CAN BE CLEARED
684 MOV #BIT0,TEMP ;LOAD INITIAL BIT
685 MOV #BITDAT|BIT4|BITS,@ICSR ;LOAD INHIBIT INPUT AND INVERT DATA
686 MOV TEMP,ACSR ;LOAD EXPECTED
687 COM ACSR ;MAKE OPPOSITE
688 MOV @DIR,R0 ;READ INPUT
689 MOV TEMP,@DIR ;CLEAR THE INPUT BIT
690 BIC #BITS,@ICSR ;REM INVERT DATA BITOVE
691 BIS #BIT1,@ICSR ;ENABLE EXT, STROBE TO PREVENT DATA INPUT BEING
692 MOV @DIR,ASTAT ;READ INPUT REG.
693 CMP ACSR,ASTAT ;COMPARE
694 BEQ 2$ ;BR IF SAME
695 ;*****
696 HRDR#,BEGIN,NULL ;INPUT REGISTER BIT FAILED TO CLEAR
697 ;*****
698
699 281 BREAK#,BEGIN ;TEMPORARY RETURN TO MONITOR,...
700 003264# 104400# 000000# ;THEN CONTINUE AT NEXT INSTRUCTION.
701 003270# 104400# 000000# ;SHIFT THE DATA
702 003274# 006367# 174772# ;TRY MORE BITS
703 003300# 001334#

```

```
703  
704  
705 003302 012777 010062 174730 DI27: ;VERIFY THAT A 2ND STROBE PULSE WILL NOT CHANGE THE DIR DATA  
706 003310 052777 004200 174722 MOV #BIT12|BIT5|BIT4|BIT1,@ICSR ;DISABLE INPUTS, ENABLE INVERT DATA, EXT  
707 003316 042777 000040 174714 BIS #BITEXT|BIT7,@ICSR ;GENERATE MAINT. STROBE  
708 003320 052777 004200 174706 BIC #BITS,@ICSR ;REMOVE INVERT DATA  
709 003332 012767 177777 174542 BIS #BITEXT|BIT7,@ICSR ;SET MAINT. STROBE AGAIN  
710 003340 017767 174700 174536 MOV #-1,@CSR ;LOAD EXPECTED DATA  
711 003346 026767 174530 174530 MOV @DIR,ASTAT ;READ REGISTER  
712 003354 001403 CMP ACSR,ASTAT ;COMPARE  
713 BEQ D130 ;BR IF SAME  
714 003356 104405 000000 000000 ;*****  
715 HRDRS,BEGIN,NULL ;DATA READY FAILED TO INHIBIT 2ND  
716 ;*****  
717 ;STROBE FROM CHAINING THE DIR  
718 003364 012777 003430 174666 DI30: ;INTERRUPT TEST -- VERIFY MNCDI INTERRUPTS VIA DATA READY VECTOR  
719 003372 116777 174414 174662 MOV #1,@DIDINV ;LOAD RETURN VECTOR  
720 003400 012777 000102 174632 MOVBR BR1,@DIDINS ;LOAD RETURN LEVEL  
721 003406 052777 004200 174624 MOV #BIT6|BIT1,@ICSR ;SET STROBE MODE  
722 003414 000240 BIS #BITEXT|BIT7,@ICSR ;GENERATE MAINT. STROBE  
723 003416 000240 NOP  
724 003420 000240 NOP  
725 003422 000240 NOP  
726 003424 104400 000000 EXIT$,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,  
727  
728 003430 005077 174604 18: CLR @ICSR  
729  
730 003434 000004 000000 003442 PIRQ$,BEGIN,2$ ; QUEUE UP TO CONTINUE AT 2$ AND RTI  
731  
732 003442 012777 000262 174610 28: MOV #DIDINS,@DIDINV  
733 003450 005077 174606 CLR @DIDINS  
734  
735  
736  
737 003454 012777 003524 174602 DI31: ;INTERRUPT TEST -- VERIFY MNCDI INTERRUPTS VIA OVERRUN ERROR  
738 003462 116777 174324 174576 MOV #1,@DIEINV ;LOAD RETURN VECTOR  
739 003470 012777 040002 174542 MOVBR BR1,@DIEINS ;LOAD RETURN STATUS  
740 003476 052777 104200 174534 BIS #BIT14|BIT1,@ICSR ;ENABLE INTR. AND STROBE MODE  
741 003504 052777 104200 174526 BIS #BIT15|BITEXT|BIT7,@ICSR ;GENERATE MAINT. STROBE  
742 003512 000240 ;GENERATE MAINT. STROBE AGAIN TO SET OVE  
743 003514 000240 NOP  
744 003516 000240 NOP  
745 003520 104400 000000 EXIT$,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,  
746 003524 005077 174510 18: CLR @ICSR  
747  
748 003530 000004 000000 003536 PIRQ$,BEGIN,2$ ; QUEUE UP TO CONTINUE AT 2$ AND RTI  
749  
750 003536 012777 000266 174520 28: MOV #DIEINS,@DIEINV  
751 003544 005077 174516 CLR @DIEINS  
752 003550 036767 174522 BIT TEMP2,SRMINE ;TEST IF INPUT ONLY  
753 003556 001002 BNE D00 ;BR IF WRAP-AROUND MODE  
754 003560 000167 174634 JMP CONT2 ;NO -- TRY NEXT UNIT
```

```
755 ;VERIFY CORRECT MNCDO ADDRESS RESPONSE  
756 003564 016767 174440 174306 D00: MOV OCSR,CSRA ;LOAD BUS ADDRESS  
757 003572 005777 174432 TST @OCSR ;TEST OUTPUT STATUS REGISTER  
758 003576 005777 174432 TST @DOR ;TEST OUTPUT DATA REGISTER  
759 003602  
760 D01: ;FLOAT A 1 ACROSS THE MNCDO DATA REGISTER  
761 003602 012767 000001 174272 MOV #BIT0,ACSR ;LOAD EXPECT BIT  
762 003610 016777 174266 174416 18: MOV ACSR,@DOR ;LOAD MNCDO DATA REGISTER  
763 003616 017767 174412 174260 MOV @DOR,ASTAT ;READ MNCDO DATA REGISTER  
764 003624 026767 174252 174252 CMP ACSR,ASTAT ;COMPARE  
765 003632 001403 BEQ 2$ ;BR IF SAME  
766  
767 003634 104405 000000 000000 ;*****  
768 HRDRS,BEGIN,NULL ;MNCDO DATA REGISTER FAILED TO HOLD A FLOATING 1  
769 ;*****  
770 003642 104407 000000 28: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR...  
771 003646 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
772 003652 006367 174224 ASL ACSR ;CHANGE THE DATA  
773 003656 001354 BNE 1$ ;BR IF MORE DATA  
774 003660  
775 D02: ;FLOAT A 0 ACROSS THE MNCDO DATA REGISTER  
776 003660 012767 000001 174404 MOV #BIT0,TEMP ;LOAD INITIAL BIT  
777 003666 016767 174400 174206 18: MOV TEMP,ACSR ;LOAD EXPECTED  
778 003674 005167 174202 COM ACSR ;COMPLEMENT  
779 003700 016777 174176 174326 MOV ACSR,@DOR ;LOAD MNCDO DATA REGISTER  
780 003706 017767 174322 174170 MOV @DOR,ASTAT ;READ MNCDO DATA REGISTER  
781 003714 026767 174162 174162 CMP ACSR,ASTAT ;COMPARE  
782 003722 001403 BEQ 2$ ;BR IF SAME  
783  
784 003724 104405 000000 000000 ;*****  
785 HRDRS,BEGIN,NULL ;MNCDO DATA REGISTER FAILED TO HOLD A FLOATING 0  
786 ;*****  
787 003732 104407 000000 28: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR...  
788 003736 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
789 003742 006367 174324 ASL TEMP ;CHANGE THE DATA  
790 003746 001347 BNE 1$ ;BR IF MORE DATA  
791 003750  
792 D03: ;VERIFY BYTE OPERATION ON THE MNCDO DATA REGISTER  
793 003750 012777 177777 174256 18: MOV #-1,@DOR ;LOAD MNCDO DATA REGISTER  
794 003756 012767 000377 174116 MOV #377,ACSR ;LOAD EXPECTED  
795 003764 105077 174246 CLRBR @DOR1 ;CLEAR HIGH BYTE  
796 003770 017767 174240 174106 MOV @DOR,ASTAT ;READ MNCDO DATA REGISTER  
797 003776 026767 174100 174100 CMP ACSR,ASTAT ;COMPARE  
798 004004 001403 BEQ 2$ ;BR IF SAME  
799  
800 004006 104405 000000 000000 ;*****  
801 HRDRS,BEGIN,NULL ;CLEARING HIGH BYTE CHANGED LOW BYTE  
802 ;*****  
802 004014 012777 177777 174212 28: MOV #-1,@DOR ;LOAD MNCDO DATA REGISTER  
803 004022 012767 177400 174052 MOV #177400,ACSR ;LOAD EXPECTED  
804 004030 105077 174200 CLRBR @DOR ;CLEAR LOW BYTE  
805 004034 017767 174174 174042 MOV @DOR,ASTAT ;READ MNCDO DATA REGISTER  
806 004042 026767 174034 174034 CMP ACSR,ASTAT ;COMPARE  
807 004050 001403 BEQ 3$ ;BR IF SAME  
808  
809 004052 104405 000000 000000 ;*****  
810 HRDRS,BEGIN,NULL ;CLEARING LOW BYTE CHANGED HIGH BYTE  
811 ;*****
```

```
011 004060*
012 004060* 104407 000000*
013 004064* 104407 000000*
014 004070*
015
016 004070* 012767 000100 174004
017 004076* 016777 174000 174124
018 004104* 017767 174120 173772
019 004112* 042767 000200 173764
020 004120* 026767 173756 173756
021 004126* 001403
022
023 004130* 104405 000000* 000000
024
025 004136* 046777 173740 174064
026 004144* 017767 174060 173732
027 004152* 026767 173724 173724
028 004160* 001003
029
030 004162* 104405 000000* 000000
031
032 004170*
033 004170* 104407 000000*
034 004174* 104407 000000*
035 004200*
036
037 004200* 012767 000020 173674
038 004206* 016777 173670 174014
039 004214* 017767 174010 173662
040 004222* 042767 000200 173654
041 004230* 026767 173646 173646
042 004236* 001403
043
044 004240* 104405 000000* 000000
045
046 004246* 046777 173630 173754
047 004254* 017767 173750 173622
048 004262* 026767 173614 173614
049 004270* 001003
050
051 004272* 104405 000000* 000000
052
053 004300*
054 004300* 104407 000000*
055 004304* 104407 000000*

38i
BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.

D04i
;TEST THAT BIT6 OF MNCDO STATUS REGISTER IS READ-WRITE
MOV #BIT6,ACSR ;LOAD EXPECTED
MOV ACSR,ROCSR ;LOAD BIT6 INTO MNCDO STATUS REGISTER
MOV ROCSR,ASTAT ;READ MNCDO STATUS REGISTER
BIC #BIT7,ASTAT ;CLEAR BIT 7
CMP ACSR,ASTAT ;TEST THAT IT SET
BEQ 16 ;BR IF SAME
;*****
HRDR$,BEGIN,NULL ;BIT6 OF MNCDO STATUS REGISTER FAILED TO SET
;*****
BIC ACSR,ROCSR ;CLEAR THAT BIT
MOV ROCSR,ASTAT ;READ MNCDO STATUS REGISTER AGAIN
CMP ACSR,ASTAT ;TEST THE BIT
BNE 26 ;BR IF CLEARED
;*****
HRDR$,BEGIN,NULL ;BIT6 OF MNCDO STATUS REGISTER FAILED TO CLEAR
;*****

26i
BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.

D05i
;TEST THAT BIT4 OF MNCDO STATUS REGISTER IS READ-WRITE
MOV #BIT4,ACSR ;LOAD EXPECTED
MOV ACSR,ROCSR ;LOAD BIT4 INTO MNCDO STATUS REGISTER
MOV ROCSR,ASTAT ;READ MNCDO STATUS REGISTER
BIC #BIT7,ASTAT ;CLEAR BIT 7
CMP ACSR,ASTAT ;TEST THAT IT SET
BEQ 16 ;BR IF SAME
;*****
HRDR$,BEGIN,NULL ;BIT4 OF MNCDO STATUS REGISTER FAILED TO SET
;*****
BIC ACSR,ROCSR ;CLEAR THAT BIT
MOV ROCSR,ASTAT ;READ MNCDO STATUS REGISTER AGAIN
CMP ACSR,ASTAT ;TEST THE BIT
BNE 26 ;BR IF CLEARED
;*****
HRDR$,BEGIN,NULL ;BIT4 OF MNCDO STATUS REGISTER FAILED TO CLEAR
;*****

28i
BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
```

```
856 004310*
857
858 004310* 012767 000010 173564
859 004316* 016777 173560 173704
860 004324* 017767 173700 173552
861 004332* 042767 000200 173544
862 004340* 026767 173536 173536
863 004346* 001403
864
865 004350* 104405 000000* 000000
866
867 004356* 046777 173520 173644
868 004364* 017767 173640 173512
869 004372* 026767 173504 173504
870 004400* 001003
871
872 004402* 104405 000000* 000000
873
874 004410*
875 004410* 104407 000000*
876 004414* 104407 000000*
877
878 004420* 005077 173604
879 004424* 012767 000200 173450
880 004432* 105077 173576
881 004436* 112777 000001 173566
882 004444* 017767 173560 173432
883 004452* 026767 173424 173424
884 004460* 001403
885
886 004462* 104405 000000* 000000
887
888
889 004470* 105077 173540
890 004474* 112777 000001 173530
891 004502* 005067 173374
892 004506* 005077 173516
893 004512* 017767 173512 173364
894 004520* 001403
895
896 004522* 104405 000000* 000000
897
898
899 004530* 105077 173500
900 004534* 112777 000001 173470
901 004542* 005067 173334
902 004546* 005077 173462
903 004552* 017767 173452 173324
904 004560* 001403
905
906 004562* 104405 000000* 000000
907

D06i
;TEST THAT BIT3 OF MNCDO STATUS REGISTER IS READ-WRITE
MOV #BIT3,ACSR ;LOAD EXPECTED
MOV ACSR,ROCSR ;LOAD BIT3 INTO MNCDO STATUS REGISTER
MOV ROCSR,ASTAT ;READ MNCDO STATUS REGISTER
BIC #BIT7,ASTAT ;CLEAR BIT 7
CMP ACSR,ASTAT ;TEST THAT IT SET
BEQ 16 ;BR IF SAME
;*****
HRDR$,BEGIN,NULL ;BIT3 OF MNCDO STATUS REGISTER FAILED TO SET
;*****
BIC ACSR,ROCSR ;CLEAR THAT BIT
MOV ROCSR,ASTAT ;READ MNCDO STATUS REGISTER AGAIN
CMP ACSR,ASTAT ;TEST THE BIT
BNE 26 ;BR IF CLEARED
;*****
HRDR$,BEGIN,NULL ;BIT3 OF MNCDO STATUS REGISTER FAILED TO CLEAR
;*****

26i
BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
;VERIFY THAT MNCDO DONE FLAG SETS

D07i
CLR ROCSR ;CLEAR CLEARED FLAG
MOV #BIT7,ACSR ;LOAD EXPECTED
CLRB ODOR ;ENABLE
MOVV #BIT0,ROCSR1 ;GENERATE MAINT. REPLY
MOV ROCSR,ASTAT ;READ OUTPUT STATUS REGISTER
CMP ACSR,ASTAT ;COMPARE
BEQ D010 ;BR IF SAME
;*****
HRDR$,BEGIN,NULL ;OUTPUT DONE FLAG FAILED TO SET
;*****
;VERIFY THAT MNCDO DONE FLAG CLEARS WHEN WRITTEN TO A 0

D010i
CLRB ODOR ;ENABLE
MOVV #BIT0,ROCSR1 ;GENERATE MAINT. REPLY
CLR ACSR ;CLEAR EXPECTED
CLR ROCSR ;CLEAR OUTPUT DONE FLAG
MOV ROCSR,ASTAT ;READ STATUS
BEQ D011 ;BR IF SAME
;*****
HRDR$,BEGIN,NULL ;CLEARING THE OUTPUT FLAG FAILED TO CLEAR OUTPUT DONE FL
;*****
;VERIFY THAT MNCDO DONE FLAG CLEARS WHEN OUTPUT DATA REGISTER IS WRITTEN

D011i
CLRB ODOR ;ENABLE
MOVV #BIT0,ROCSR1 ;GENERATE MAINT. REPLY
CLR ACSR ;CLEAR EXPECTED
CLR ODOR ;WRITE THE OUTPUT DATA REGISTER
MOV ROCSR,ASTAT ;READ OUTPUT STATUS REGISTER
BEQ D012 ;BR IF CLEARED
;*****
HRDR$,BEGIN,NULL ;OUTPUT DONE FLAG FAILED TO CLEAR WHEN THE DOR REGISTER
;*****
```

```
908 ;INTERRUPT TEST == VERIFY MNCDO DOES INTERRUPT
909 004570 012777 004636 173456 DO12: MOV #1,@DODINV ;LOAD RETURN VECTOR
910 004576 116777 173210 173452 MOVB BRI,@DODINS ;LOAD RETURN LEVEL
911 004604 105077 173424 CLRB @DOR ;ENABLE
912 004610 112777 000001 173414 MOVB #BIT0,@OCSR1 ;GENERATE MAINT, REPLY
913 004616 052777 000100 173404 BIS #BIT6,@OCSR ;ENABLE INTR.
914 004624 000240 NOP
915 004626 000240 NOP
916 004630 000240 NOP
917 004632 104400 000000 EXIT$,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,
918
919 004636 005077 173366 10: CLR @OCSR
920 ;-----
921 004642 000004 000000 004650 PIRQ$,BEGIN,36 ; QUEUE UP TO CONTINUE AT 36 AND RTI
922 ;-----
923 004650 016777 173402 173376 30: MOV DODINS,@DODINV ;RESET VECTORS
924 004656 005077 173374 CLRB @DODINS
925 004662 005077 173342 CLR @OCSR
```

```
926 ;WRAP AROUND TESTS
927
928 ;VERIFY MODE 00 -- INPUT STROBE WILL SET THE INPUT DATA READY FLAG
929 004666 012777 177777 173350 DI32: MOV #-1,@DIR ;CLEAR INPUT REG.
930 004674 005077 173350 CLRB @SBR ;CLEAR STIM, REG.
931 004700 005077 173334 CLRB @ICSR ;CLEAR INPUT DATA READY FLAG
932 004704 005077 173320 CLRB @OCSR ;CLEAR OUTPUT STATUS
933 004710 012777 025252 173316 MOV #25252,@DOR ;WRITE TO THE OUTPUT DATA REG.
934 004716 012767 000200 173156 MOV #BIT7,@ACSR ;LOAD EXPECTED
935 004724 017767 173310 173152 MOV @ICSR,ASTAT ;READ STATUS
936 004732 026767 173144 173144 CMP ACSR,ASTAT ;COMPARE
937 004740 001406 BEQ DI33 ;BR IF SAME
938
939 004742 104405 000000 000000 HRDR$,BEGIN,NULL ;MODE 00 -- EXT, STROBE FAILED TO SET INPUT DATA READY
940 ;-----
941 004750 104403 000000 006030 MSGN$,BEGIN,MSG1 ;ASCII MESSAGE CALL WITH COMMON HEADER
942
943 ;VERIFY MODE 01 -- INPUT STROBE WILL SET THE INPUT DATA READY FLAG
944 004756 012777 177777 173260 DI33: MOV #-1,@DIR ;CLEAR INPUT REG.
945 004764 005077 173260 CLRB @SBR ;CLEAR STIM, REG.
946 004770 012777 000002 173242 MOV #BIT1,@ICSR ;CLEAR INPUT DATA READY FLAG
947 004776 005077 173226 CLRB @OCSR ;CLEAR OUTPUT STATUS
948 005002 012777 052525 173224 MOV #52525,@DOR ;WRITE TO THE OUTPUT DATA REG.
949 005010 012767 000202 173064 MOV #BIT7|BIT1,ACSR ;LOAD EXPECTED
950 005016 017767 173216 173060 MOV @ICSR,ASTAT ;READ STATUS
951 005024 026767 173052 173052 CMP ACSR,ASTAT ;COMPARE
952 005032 001403 BEQ DI34 ;BR IF SAME
953
954 005034 104405 000000 000000 HRDR$,BEGIN,NULL ;MODE 01 -- EXT, STROBE FAILED TO SET INPUT DATA READY
955 ;-----
956
957 ;VERIFY MODE 10 -- INPUT STROBE WILL NOT SET INPUT DATA READY FLAG
958 005042 012777 177777 173174 DI34: MOV #-1,@DIR ;CLEAR INPUT REG.
959 005050 005077 173174 CLRB @SBR ;CLEAR STIM, REG.
960 005054 012777 000004 173156 MOV #BIT2,@ICSR ;CLEAR INPUT DATA READY FLAG
961 005062 005077 173142 CLRB @OCSR ;CLEAR OUTPUT STATUS
962 005066 012777 070707 173140 MOV #70707,@DOR ;WRITE TO THE OUTPUT DATA REG.
963 005074 012767 000004 173000 MOV #BIT2,ACSR ;LOAD EXPECTED
964 005102 017767 173132 172774 MOV @ICSR,ASTAT ;READ STATUS
965 005110 026767 172766 172766 CMP ACSR,ASTAT ;COMPARE
966 005116 001403 BEQ DI35 ;BR IF SAME
967
968 005120 104405 000000 000000 HRDR$,BEGIN,NULL ;MODE 10 -- EXT, STROBE SET INPUT DATA READY FLAG IN ER
969 ;-----
```

```

970 ;VERIFY INPUT REPLY SETS OUTPUT DONE FLAG
971 005126 005077 173102 DI35: CLR @DOR ;CLEAR OUTPUT DATA
972 005132 012767 000200 172742 MOV #BIT7,ACSR ;LOAD EXPECTED
973 005140 005077 173064 CLR @OCSR ;CLEAR OUTPUT DONE FLAG
974 005144 012777 004200 173066 MOV #BITEXT|BIT7,@ICSR ;SET INPUT READY FLAG
975 005152 005077 173062 CLR @ICSR ;CLEAR INPUT READY FLAG<GEN. INPUT REPLY>
976 005156 017767 173046 172720 MOV @OCSR,ASTAT ;READ OUTPUT STATUS
977 005164 026767 172712 172712 CMP ACSR,ASTAT ;COMPARE
978 005172 001403 BEQ DI36 ;BR IF SAME
979
980 005174 104405 000000 000000 ;*****
981 HRDR#,BEGIN,NULL ;INPUT REPLY FAILED TO SET OUTPUT DONE FLAG
982 ;*****
983 005202 012777 177777 173034 DI36: MOV #-1,@DIR ;CLEAR INPUT REG.
984 005210 005077 173014 CLR @OCSR ;CLEAR OUTPUT STATUS
985 005214 005077 173020 CLR @ICSR ;CLEAR INPUT STATUS
986 005220 012767 000001 173044 MOV #BIT0,TEMP ;LOAD EXPECTED
987 005226 016767 173040 172646 MOV TEMP,ACSR ;LOAD TYPEOUT EXPECTED
988 005234 016777 172642 172772 MOV ACSR,@DOR ;LOAD OUTPUT DATA REG.
989 005242 017767 172776 172634 MOV @DIR,ASTAT ;READ INPUT DATA REGISTER
990 005250 026767 172626 172626 CMP ACSR,ASTAT ;COMPARE
991 005256 001403 BEQ 28 ;BR IF SAME
992 ;*****
993 005260 104405 000000 000000 HRDR#,BEGIN,NULL ;INPUT DATA PATH ERROR
994 ;*****
995 005266 28:
996 005266 104407 000000 BREAK#,BEGIN ;TEMPORARY RETURN TO MONITOR,...
997 005272 104407 000000 BREAK#,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
998 005276 006367 172770 ASL TEMP ;TRY NEXT BIT
999 005302 001351 BNE 18 ;BR IF MORE BITS
1000 ;VERIFY THE MNCDO - WRAPAROUND - MNCDI INVERTED DATA PATH
1001 005304 012777 177777 172732 DI37: MOV #-1,@DIR ;CLEAR INPUT REG.
1002 005312 005077 172712 CLR @OCSR ;CLEAR OUTPUT STATUS
1003 005316 012777 000060 172714 MOV #BIT5|BIT4,@ICSR ;LOAD INPUT STATUS <INVERT DATA>
1004 005324 012767 000001 172740 MOV #BIT0,TEMP ;LOAD INITIAL BIT
1005 005332 016777 172734 172674 18: MOV TEMP,@DOR ;LOAD OUTPUT DATA REG.
1006 005340 017767 172700 172536 MOV @DIR,ASTAT ;READ INPUT DATA REGISTER
1007 005346 016767 172720 172526 MOV TEMP,ACSR ;GET THE BIT
1008 005354 005167 172522 COM ACSR ;INVERT EXPECTED INPUT DATA
1009 005360 026767 172516 172516 CMP ACSR,ASTAT ;COMPARE
1010 005366 001403 BEQ 28 ;BR IF SAME
1011 ;*****
1012 005370 104405 000000 000000 HRDR#,BEGIN,NULL ;INVERTED INPUT DATA PATH ERROR
1013 ;*****
1014 005376 28:
1015 005376 104407 000000 BREAK#,BEGIN ;TEMPORARY RETURN TO MONITOR,...
1016 005402 104407 000000 BREAK#,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1017 005406 006367 172660 ASL TEMP ;TRY NEXT BIT
1018 005412 001347 BNE 18 ;BR IF MORE BITS
1019

```

```

1020 ;VERIFY IN 10 MODE THAT SBR AND INPUT BITS SET INPUT READY
1021 ; INPUT DATA READY FLAG
1022 ;
1023 ; LOAD A FLOATING 1 ACROSS THE SBR
1024 ; VERIFY THAT ONLY THE CORRECT BIT SET DATA READY
1025 005414 005077 172614 DI40: CLR @DOR
1026 005420 012767 000001 172644 MOV #BIT0,TEMP ;LOAD INITIAL BIT
1027
1028 005426 005077 172602 18: CLR @DOR ;CLEAR OUTPUT BITS
1029 005432 005077 172612 CLR @SBR ;CLEAR SBR REG.
1030 005436 012777 000004 172574 MOV #BIT2,@ICSR ;CLEAR INPUT READY AND SET MODE 10
1031 005444 012777 177777 172572 MOV #-1,@DIR ;CLEAR INPUT REG.
1032
1033 005452 016777 172614 172570 MOV TEMP,@SBR ;LOAD SBR REG.
1034 005460 042777 000200 172552 BIC #BIT7,@ICSR ;CLEAR INPUT READY BIT
1035 005466 016777 172600 172540 MOV TEMP,@DOR ;LOAD OUTPUT REG.
1036
1037 005474 012767 100204 172400 MOV #BIT15|BIT7|BIT2,ACSR ;LOAD EXPECTED STATUS
1038 005502 017767 172532 172374 MOV @ICSR,ASTAT ;READ STATUS
1039 005510 026767 172366 172366 CMP ACSR,ASTAT ;COMPARE
1040 005516 001403 BEQ 28 ;BR IF SAME
1041 ;*****
1042 005520 104405 000000 000000 HRDR#,BEGIN,NULL ;INPUT DATA READY FLAG FAILED
1043 ;*****
1044 ;TO SET IN MODE 10 <STIMULUS MODE>
1045
1046 ;NOW LOAD ALL BITS EXCEPT THE FLOATING BIT AND ENSURE INPUT DATA READY DOES NOT SET
1047 005526 016767 172540 172540 28: MOV TEMP,TEMP1 ;COPY EXPECTED
1048 005534 005167 172534 COM TEMP1 ;USE REVERSE PATTERN
1049 005540 005077 172470 CLR @DOR ;CLEAR OUTPUT REG.
1050 005544 012777 177777 172472 MOV #-1,@DIR ;CLEAR INPUT REG.
1051 005552 042777 100200 172460 BIC #BIT15|BIT7,@ICSR ;CLEAR INPUT DATA READY
1052 005560 012767 000004 172314 MOV #BIT2,ACSR ;LOAD EXPECTED
1053
1054 005566 016777 172502 172440 MOV TEMP1,@DOR ;LOAD ALL OTHER DATA BITS
1055 005574 017767 172440 172302 MOV @ICSR,ASTAT ;READ INPUT STATUS
1056 005602 026767 172274 172274 CMP ACSR,ASTAT ;COMPARE
1057 005610 001403 BEQ 38 ;BR IF SAME
1058 ;*****
1059 005612 104405 000000 000000 HRDR#,BEGIN,NULL ;INPUT DATA READY FLAG SET IN ERROR
1060 ;*****
1061 ;UNEXPECTED SBR BIT SET INPUT DATA READY
1062
1063 005620 38:
1064 005624 104407 000000 BREAK#,BEGIN ;TEMPORARY RETURN TO MONITOR,...
1065 005630 104407 000000 BREAK#,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1066 005636 006367 172436 ASL TEMP ;TRY NEXT BIT
1067 005634 001274 BNE 18 ;BR IF MORE BITS

```

```

1068 ;TEST THE TRANSITION ENABLE AND TRANSITION DETECTION
1069 005636 005077 172372 DI41: CLR @DOR
1070 005642 012777 177777 MOV #=1,@DIR ;CLEAR INPUT REGISTER
1071 005650 012767 010000 MOV #BIT12,TEMP ;LOAD INITIAL TRANSITION BIT
1072 005656 012777 000404 MOV #BIT8|BIT2,@ICSR ;SET STIM. CLEAR READY, ENABLE TRANS.
1073 005664 012767 100604 MOV #BIT15|BIT8|BIT7|BIT2,ACSR ;LOAD EXPECTED STATUS
1074 005672 042777 100200 BIC #BIT15|BIT7,@ICSR ;CLEAR READY
1075 005700 016777 172366 MOV TEMP,@SBR ;LOAD STIMULUS REG.
1076 005706 016777 172360 MOV TEMP,@DOR ;LOAD INPUT REG. <VIA OUTPUT REG.>
1077 005714 017767 172320 MOV #ICSR,ASTAT ;READ INPUT STATUS
1078 005722 026767 172154 CMP ACSR,ASTAT ;COMPARE
1079 005730 001403 BEQ 28 ;BR IF SAME
1080
1081 005732 104405 000000 HRDERS,BEGIN,NULL ;TRANSITION ENABLE OR TRANSITION TO A ONE FAILED
1082
1083
1084 ;NOW REMOVE THE TRANSITION DATA BIT (THIS SHOULD CAUSE THE INPUT READY FLAG TO SET AGAIN)
1085 005740 012777 177777 20: MOV #=1,@DIR ;CLEAR INPUT REG
1086 005746 042777 100200 BIC #BIT15|BIT7,@ICSR ;CLEAR INPUT READY FLAG
1087 005754 046777 172312 BIC TEMP,@DOR ;REMOVE THE INPUT DATA
1088
1089 005762 017767 172252 MOV #ICSR,ASTAT ;THIS SHOULD CAUSE THE TRANSITION TO A ZERO
1090 005770 026767 172106 CMP ACSR,ASTAT ;READ INPUT STATUS
1091 005776 001403 BEQ 38 ;BR IF SAME
1092
1093 006000 104405 000000 HRDERS,BEGIN,NULL ;TRANSITION TO A ZERO FAILED
1094
1095
1096 006006 30: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR,...
1097 006006 104407 000000 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1098 006012 104407 000000 ASL TEMP ;TRY ANOTHER BIT ?
1099 006016 006367 172250 BNE 18 ;BR IF YES
1100 006022 001323
1101
1102 006024 000167 172370 JMP CONT2 ;TRY NEXT UNIT
1103
1104 006030 006034 MSG1: PRI1 ;POINTER TO TEXT
1105 006032 177777 =1 ;TERMINATOR
1106 006034 053045 051105 043111 PRI1: ,ASCIZ \&VERIFY MNCDO WRAP-AROUND CONNECTION TO MNCDI\
1107 006042 020131 047115 042103
1108 006050 020117 051127 050101
1109 006056 040455 047522 047125
1110 006064 020104 047503 047116
1111 006072 041505 044524 047117
1112 006100 052040 020117 047115
1113 006106 042103 000111
1114
1115 000001 .EVEN
.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#												
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	693	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	609	622	625	626	636	646	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	963
		980	993	996	997	1012	1015	1016	1042	1059	1063	1064	1081	1098
		1097	1098											
BITDAT=	010000	241#	654	661	674	685	685	643	706	708	721	740	741	
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	406	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														
CSRA 000100R	196#	299*	756*														
DATCK# = 104411	211#																
DATER# = 104404	211#																
DIDINS 000262R	234#	719*	732	733*													
DIDINV 000260R	233#	718*	732*														
DIEINS 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	501#															
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#																

DODINS 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#																
ERRRYP 000106R	201#																
EXIT# = 104400	211#	726	745	917													
GETPA# = 104415	211#																
GWBUP# = 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*	748*	751*	765*	766*	767*	768*	782*	783*				
	98																

	784	800	809	823	830	844	851	865	872	886	896	906	939
OCSR	000230R	954	968	980	993	1012	1042	1059	1081	1093			
		216*	246	272	756	757	817*	818	825*	826	838*	839	846*
		859*	860	867*	868	878*	882	892*	893	903	913*	919*	925*
		947*	961*	973*	976	984*	1002*						
OCSR1	000232R	217*	881*	890*	900*	912*							
OPEN	= 000000	153	169	170	171	172	189	190	191	192	193	194	195
		198	200	202	203	205	206	207	211*				
OTOA8	= 104420	211*											
PASCNT	000034R	177*											
PIRO8	= 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*											
RAND8	= 104417	211*											
RANNUM	000054R	185*											
RESTR	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*
		945*	959*	1029*	1033*	1075*							930*
SBR1	000252R	228*	339*										
SOFcnt	000042R	180*											
SOFER8	= 104406	211*											
SOFPAS	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*											

SYSCT	000052R	184*											
TEMP	000272R	238*	320*	321	333*	684*	686	689	701*	776*	777	789*	986*
		998*	1004*	1005	1007	1017*	1026*	1033	1035	1047	1065*	1071*	1075
		1087	1099*										1076
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)