

•REM

IDENTIFICATION

PRODUCT CODE: AC-F234A-MC
PRODUCT NAME: CXNCBA0 NCV-11A MODULE
PRODUCT DATE: FEB 1979
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) 1978,1979 DIGITAL EQUIPMENT CORPORATION

1. PAGE

ABSTRACT

"NCR" IS AN "BIOMODX" THAT EXERCISES ONE NCV-11A INTERFACE.
THE NCV-11A INTERFACE DOES MEMORY INCREMENTS VIA NPR UNTIL
A WORD OR BYTE REACHES MAXIMUM CAPACITY AND ATTEMPTS TO
OVERFLOW. AT THIS TIME AN INTERRUPT IS GENERATED AT PR LEVEL 6.
THE INTERFACE ALSO DOES TRANSFERS OF DATA TO SERIAL LOCATIONS
IN CORE VIA NPR. THIS MODE IS TERMINATED BY A WORD COUNT OVERFLOW
AND CONSEQUENT INTERRUPT. THE RATE OF INCREMENT OR TRANSFER IS SET
BY A CLOCK SIGNAL DEVELOPED ON THE NCV11 CONTROLLER.

2. REQUIREMENTS

HARDWARE: NCV-11A INTERFACE
STORAGE: NCR MODULE REQUIRES 1400. WORDS OF STORAGE

3. PASS DEFINITION

ONE PASS OF NCA MODULE CONSISTS OF EIGHTY ITERATIONS OF EACH BASIC TEST
SEQUENCE, WHICH RESULTS IN:

200 PROGRAM INTERRUPTS, 87,000 NON-PROCESSOR REQUESTS.

4. EXECUTION TIME

NCB RUNNING ALONE ON PDP-11/34 TAKES APPROXIMATELY 60 SECONDS.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172760, VECTOR: 370, BRI: 6, DEVCNT: N/A

REQUIRED PARAMETERS:

NONE

DEVICE/OPTION SETUP

NONE.

7. MODULE OPERATION

THE MODULE CONSISTS OF A LOGIC TEST, LIST MODE, AND MATRIX
MODE DATA TRANSFERS, THE LOGIC TEST PROVIDES A QUICK TEST OF
THE MAJOR REGISTER FOR OPERATION. THE LIST MODE TRANSFERS
ARE FIRST EXECUTED IN MAINTENANCE MODE AND AT FULL SPEED. THE
MODULE WILL THEN COLLECT DATA IN MATRIX MODE AT FULL SPEED.
THE SEQUENCE IS REPEATED UNTIL THE PASS COUNTER IS EXHAUSTED.
UPON COMPLETION, AN END OF PASS IS REPORTED AND THE MODULE IS
RESTARTED.

8. OPERATION OPTIONS

SRI IS USED TO INHIBIT TESTING MODES OF OPERATION OF THE NCV11A.
IF THE BITS ARE USED THE INTERACTION COUNT LOCATION "PASS" MUST BE
MODIFIED TO EXTEND THE EXECUTION TIME BEFORE THE END OF PASS REPORT.

SRI BIT0 = 1 INHIBIT MATRIX WORD INCREMENT MODE.
SRI BIT1 = 1 INHIBIT LIST MODE.

9. NON STANDARD PRINTOUTS

NONE. ALL PRINTOUTS HAVE STANDARD MEANINGS AS REPRESENTED IN
DEC/x11 DOCUMENTATION.

10. MODULE TEST ENVIRONMENT

THE NCA MODULE IS KNOWN TO OPERATE UNDER THIS ENVIRONMENT:

#1	PDP-11/34 CPU WITH 64K	#2	PDP-11/40 CPU WITH 28K
	TC11 2 DRIVES		RK11-D 1 DRIVE
	TM11 1 DRIVE		TM11 1 DRIVE
	VSV01 1 SCOPE		NCV11A 1 UNIT
	NCV11A 1 UNIT		KW11 1 UNIT
	LP11 1 UNIT		
#3	PDP-11/34 CPU WITH 32K		
	RK11-D 1 DRIVE		
	NCV11A 1 UNIT		
	KW11L 1 UNIT		

143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195

```
.LIST SEQ,RIN  
.ELIST WC,CD,CND  
.LIST WC  
.TITLE NCRA DEC/Y11 SYSTEM EXERCISER MODULE  
.RXCOM VERSION 6 21-MAY-78  
.LIST RIN  
*****  
;BEGIN:*****  
;MODULE NAME  
;USED TO KEEP TRACK OF WBUF USAGE  
;LIST DEVICE ADDR.  
;LIST DEVICE VECTOR.  
;1ST RR LEVEL:  
;2ND RR LEVEL:  
;DEVICE INDICATOR 1.  
;SWITCH REGISTER 1  
;SWITCH REGISTER 2  
;SWITCH REGISTER 3  
;SWITCH REGISTER 4  
*****  
;STAT'S WORD.  
;MODULE START ADDR.  
;MODULE STACK POINTER.  
;PASS COUNTER.  
;# OF ITERATIONS PER PASS=80.  
;LOC TO COUNT ITERATIONS  
;LOC TO SAVE TOTAL SOFT ERRORS  
;LOC TO SAVE TOTAL HARD ERRORS  
;LOC TO SAVE SOFT ERRORS PER PASS  
;LOC TO SAVE HARD ERRORS PER PASS  
;# OF SYS ERRORS ACCUMULATED  
;HOLDS RANDOM # WHEN RAND MACRO IS CALLED  
;RESERVED FOR MONITOR USE  
;RESERVED FOR MONITOR USE  
;RESERVED FOR MONITOR USE  
;LOC TO SAVE R0.  
;LOC TO SAVE R1.  
;LOC TO SAVE R2.  
;LOC TO SAVE R3.  
;LOC TO SAVE R4.  
;LOC TO SAVE R5.  
;LOC TO SAVE R6.  
;ADDR OF CURRENT CSR.  
;ADDR OF GOOD DATA, OR  
;CONTENTS OF CSR.  
;ADDR OF BAD DATA, OR  
;STATUS REG CONTENTS.  
;TYPE OF ERROR.  
;EXPECTED DATA.  
;ACTUAL DATA.  
;RESTART ADDRESS AFTER END OF PASS  
;WORDS TO MEMORY PER ITERATION  
;WORDS FROM MEMORY PER ITERATION  
;# OF INTERRUPTS PER ITERATION  
;MODULE IDENTIFICATION NUMBER=0
```

196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251

```
RRRWA: RBUF  
RRRPA: OPEN  
RRRPS: OPEN  
WRRPA: OPEN  
WRRFR: OPEN  
WRRFS: OPEN  
CDRECT: OPEN  
CNDCT: OPEN  
PREP: OPEN  
WDOSP: OPEN  
*****  
;READ BUFFER VIRTUAL ADDRESS  
;READ BUFFER PHYSICAL ADDRESS  
;READ BUFFER EA BITS  
;SIZE OF THE READ BUFFER  
;WRITE BUFFER PHYSICAL ADDRESS  
;WRITE BUFFER EA BITS  
;WRITE BUFFER SIZE REQUESTED  
;WRITE BUFFER SIZE AVAILABLE  
;CDATA/DATCK ERROR COUNT  
;CDATA/DATCK WORD COUNT  
;RESERVED FOR FUTURE USE  
*****  
;TYPE  
;BASE=0  
;AVECT1=0  
;TSTDMA=BIT3  
;SNDMA=BIT8  
;TSTZ=BIT1  
;TSTCON=BIT2  
;TSTDMA=BIT3  
;CLRALL=BIT11  
;CSR:  
;ARASE: ARASE+2  
;WCR: ARASE+4  
;PAR: ARASE+6  
;SRP: ARASE+10  
;ADM: ARASE+12  
;JW: ARASE+14  
;RBP1: ARASE+16  
;CSRHB: ARASE+1  
;SFRH: ARASE+11  
;VECTA0: AVECT1  
;VECTA1: AVECT1+2  
;VECTR0: AVECT1+4  
;VECTR1: AVECT1+6  
;STEP: C  
;STAT: C  
;RSTRT: C  
*****  
LOGIC: MOV CLRALL,RSFP ;CLEAR NCV11  
MOV MCSR,R1 ;LOAD ADDRESS POINTER  
MOV ADDR,R1 ;GET RUS ADDRESS  
16: ADD R1,(R0)+ ;LOAD DEVICE ADDRESS  
CMP PC,#CSRHR ;UPDATE VALUE  
JNE 1S ;TEST IF DONE  
MOV ADDR,(R0) ;LOAD BYTE ADDRESSES  
MOV ADDR,(R0) ;LOAD BYTE OF SFR  
26: MOV R1,(R0)+ ;LOAD VECTORS  
CMP #VECTR1+2,R0 ;UPDATE  
JNE 2C ;RR UNTIL DONE  
MOV ADDR,CSRA ;LOAD FOR TYPE OUT
```

```
252 ;*****  
253 ;VERIFY THE H NCV11 BUS ADDRESSES RESPOND  
254 ;*****  
255 TST1: ;*****  
256 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
257 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
258 TST #QCSR ;ADDRESS  
259 TST #QDF ;  
260 TST #WCP ;THE  
261 TST #QAP ;  
262 TST #SFR ;NCV11  
263 TST #ADM ;ADDRESSES  
264 TST #RRI ;  
265 ;*****  
266 ;FLOAT A 1 ACROSS 10 BITS OF THE COMMAND/STATUS REG.  
267 ;*****  
268 TST2: ;*****  
269 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
270 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
271 MOV #RIT1,STEMP ;LOAD INITIAL REG. VALUE  
272 MOV #STEMP,#CSR ;LOAD CSR REG.  
273 MOV #CSR,#ASTAT ;READ CSR  
274 MOV #STEMP,#CSR ;LOAD EXPECTED  
275 BIS #RIT7,#CSR ;FORCE THE "READY" BIT  
276 CMP #CSR,#ASTAT ;COMPARE THE VALUES  
277 REQ #2S ;//RR IF SAME  
278 ;*****  
279 #RDERS,REGIN,NULL ;UNEXPECTED VALUE IN THE CSR REGISTER  
280 ;*****  
281 TST3: ;*****  
282 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
283 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
284 ASP #STEMP ;TRY THE NEXT DATA BIT  
285 CMP #1,STEMP ;TEST IF NOW BIT 0  
286 RNE #1S ;RR IF NOT  
287 ;*****  
288 ;VERIFY THAT "CLR ALL" CLEARS THE CSR REGISTER  
289 ;*****  
290 TST3: ;*****  
291 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
292 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
293 MOV #R776,#CSR ;LOAD CSR REG.  
294 MOV #CLRALL,#SFP ;GENERATE "CLR ALL L"  
295 MOV #RIT7,#CSR ;LOAD EXPECTED VALUE  
296 MOV #CSR,#ASTAT ;READ THE CSR REG.  
297 CMP #CSR,#ASTAT ;COMPARE VALUES  
298 REQ #2S ;//RR IF SAME  
299 ;*****  
300 #RDERS,REGIN,NULL ;"CLR ALL L" FAILED TO CLEAR CSR REG.  
301 ;*****
```

```
302 ;*****  
303 ;VERIFY LOW BYTE OPERATION OF THE "CSR" REGISTER  
304 ;*****  
305 TST4: ;*****  
306 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
307 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
308 MOV #3636,#CSR ;LOAD CSR REGISTER  
309 MOV #3600,#CSR ;LOAD EXPECTED VALUE  
310 CLR #CSR ;CLEAR LOW BYTE  
311 MOV #CSR,#ASTAT ;READ STATUS REG.  
312 CMP #CSR,#ASTAT ;COMPARE VALUES  
313 REQ #1S ;//RR IF SAME  
314 ;*****  
315 #RDERS,REGIN,NULL ;CLEARING LOW BYTE OF THE CSR CHANGED THE HIGH BYTE  
316 ;*****  
317 TST5: ;*****  
318 MOV #3636,#CSR ;LOAD CSR REGISTER  
319 MOV #236,#CSR ;LOAD EXPECTED VALUE  
320 CLR #CSR ;CLEAR HIGH BYTE OF THE CSR  
321 MOV #CSR,#ASTAT ;READ STATUS  
322 CMP #CSR,#ASTAT ;COMPARE VALUES  
323 REQ #2S ;//RR IS SAME  
324 ;*****  
325 #RDERS,REGIN,NULL ;CLEARING HIGH BYTE OF CSR CHANGED THE LOW BYTE  
326 ;*****  
327 ;FLOAT A 1 ACROSS 4 BITS OF THE SPECIAL FUNCTION REGISTER  
328 ;*****  
329 TST5: ;*****  
330 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
331 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
332 MOV #RIT4,STEMP ;LOAD INITIAL REG. VALUE  
333 ;*****  
334 TST6: ;*****  
335 MOV #STEMP,#SFR ;LOAD SFR REG.  
336 MOV #SFR,#ASTAT ;READ SFR  
337 MOV #STEMP,#CSR ;LOAD EXPECTED  
338 CMP #CSR,#ASTAT ;COMPARE THE VALUES  
339 REQ #2S ;//RR IF SAME  
340 ;*****  
341 #RDERS,REGIN,NULL ;UNEXPECTED VALUE IN THE SFR REGISTER  
342 ;*****  
343 TST7: ;*****  
344 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
345 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
346 ASP #STEMP ;TRY THE NEXT DATA BIT  
347 CMP #1,STEMP ;TEST IF NOW BIT 0  
348 RNE #1S ;RR IF NOT
```

```

142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
400

```

```

;*****
;VERIFY THAT CLEARING HIGH BYTE OF SFR DOES NOT CLEAR LOW BYTE
;*****
TST6:
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
MOV #14,MSFR ;LOAD THE S.F. REGISTER
MOV #14,ACSP ;LOAD THE EXPECTED VALUE
CLR MSFR ;CLEAR HIGH BYTE OF S.F. REG.
MOV MSFR,ASTAT ;READ THE REGISTER
CMP ACSR,ASTAT ;COMPARE THE VALUES
REQ TS7 ;//RR IF SAME
;*****
;VERIFY THAT CLEARING HIGH BYTE OF CSR REG. CHANGED THE LOW BYTE AL
;*****
TST7:
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
MOV #16,MSFR ;LOAD SFR REG.
BTS #CLRALL,MSFR ;GENERATE "CLR ALL L"
CLR ACSR ;LOAD EXPECTED VALUE
MOV MSFR,ASTAT ;READ THE SFR REG.
REQ TS10 ;//RR IF SAME
;*****
;VERIFY THAT "CLR ALL L" FAILED TO CLEAR SFR REG.
;*****
TST10:
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
MOV #R10,ACSP ;LOAD EXPECTED VALUE
15:
MOV #CLRALL,MSFR ;RESET THE DEVICE
MOV ACSR,MSFR ;LOAD WORD COUNT "A"
MOV MSFR,ASTAT ;READ WORD COUNT
CMP ACSR,ASTAT ;COMPARE VALUES
REQ TS ;//RR IF SAME
;*****
;VERIFY THAT "CLR ALL L" FAILED TO CLEAR SFR REG.
;*****
TST11:
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
ASL ACSR ;CHANGE THE DATA
RNE TS ;RR IF MORE DATA TO LOAD

```

```

401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444

```

```

;*****
;FLOAT A 1 ACROSS THE BUS ADDRESS REGISTER
;*****
TST11:
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
MOV #R10,ACSR ;LOAD EXPECTED VALUE
15:
MOV #CLRALL,MSFR ;RESET THE DEVICE
MOV ACSR,MSFR ;LOAD BUS ADDRESS "A"
MOV MSFR,ASTAT ;READ BUS ADDRESS
CMP ACSR,ASTAT ;COMPARE VALUES
REQ TS ;//RR IF SAME
;*****
;VERIFY THAT "CLR ALL L" FAILED TO CLEAR SFR REG.
;*****
TST12:
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
MOV #R10,ACSP ;LOAD EXPECTED VALUE
15:
MOV #CLRALL,MSFR ;RESET THE DEVICE
MOV ACSR,MSFR ;LOAD OFFSET "A"
MOV MSFR,ASTAT ;READ OFFSET
CMP ACSR,ASTAT ;COMPARE VALUES
REQ TS ;//RR IF SAME
;*****
;VERIFY THAT "CLR ALL L" FAILED TO CLEAR SFR REG.
;*****
TST13:
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
ASL ACSR ;CHANGE THE DATA
RNE TS ;RR IF MORE DATA TO LOAD

```



```
*****  
;ENABLE A 512 WORD TRANSFER SECTION LIST MODE (WAIT MODE)  
*****  
TST16:  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
BIT #BIT1,SRI ;TEST IF INHIBIT IS SET  
MOV #CLPALL,ASFR ;CLEAR THE DEVICE  
MOV #RUF0,R0 ;LOAD BUFFER POINTER  
MOV #7070,(R0)+ ;PRESET THE BUFFER WITH DATA  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
CMP R0,#RUF1 ;TEST IF DONE  
RNE AS ;RR IF NOT  
GETPAS,REGIN,RRUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RRUFVA  
MOV #RUFEA,R0 ;GET EXTENDED ADDRESS BITS  
ASR R0  
ASR R0  
ASR R0  
ASR R0  
BIT #177774,R0 ;CLEAR OFF EXCESSIVE  
MOV #R0,R0F ;LOAD EXTENDED BITS  
MOV #512,#WCR ;SET UP 512 WORD TRANSFER  
MOV #RUFPA,#BAR ;LOAD BUS ADDRESS FOR RESULT  
MOV #TSTCON,ASFR ;ENABLE TEST CONTROL AND DMA FLOPS  
BIS #512,STEMP ;ENABLE DEVICE  
MOV #512,STEMP ;LOAD THE COUNTER  
BIS #TESTZ,ASFR ;ENABLE INT. Z  
BIC #TESTZ,ASFR ;DISABLE INT. Z  
BIS #TESTZ,ASFR ;ALLOW 1 DMA TRANSFER  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
DEC #STEMP ;FINISHED ALL WORDS?  
RNE #S ;RR UNTILL DONE  
;THE TRANSFER IS NOW COMPLETE  
CLF ACSP ;CLEAR EXPECTED  
MOV #OFF,ASTAT ;READ OFFSET REG.  
BIC #3,ASTAT ;CLEAR LOW 2 BITS  
REQ #S ;RR IF HIGH 14 BITS ARE CLEARED  
;*****  
;UNEXPECTED OFFSET REGISTER BIT SET  
;*****  
;INCORRECT BUS ADDRESS VALUE AFTER A 1 WORD TRANSFER  
;*****  
;RR IF SAME  
;*****  
;UNEXPECTED HIGH 14 BITS OF OFFSET REGISTER BIT SET  
;*****  
;RR IF SAME  
;*****
```

```
*****  
;UNEXPECTED WORD COUNT REGISTER VALUE AFTER A 1 WORD TRA  
;*****  
;*****  
;ENABLE A 512 WORD TRANSFER SECTION LIST MODE  
*****  
TST17:  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
BIT #BIT1,SRI ;TEST IF INHIBIT IS SET  
MOV #CLPALL,ASFR ;CLEAR THE DEVICE  
MOV #RUF0,R0 ;LOAD BUFFER POINTER  
MOV #7070,(R0)+ ;PRESET THE BUFFER WITH DATA  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
CMP R0,#RUF1 ;TEST IF DONE  
RNE AS ;RR IF NOT  
GETPAS,REGIN,RRUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RRUFVA  
MOV #RUFEA,R0 ;GET EXTENDED ADDRESS BITS  
ASR R0  
ASR R0  
ASR R0  
ASR R0  
BIT #177774,R0 ;CLEAR OFF THE EXCESS  
MOV #R0,R0F ;LOAD EXTENDED ADDRESS BITS  
MOV #3,#VECTA0 ;LOAD RETURN VECTOR  
MOV #R0,#VECTA1  
MOV #512,#WCR ;SET UP 512 WORD TRANSFER  
MOV #RUFPA,#BAR ;LOAD BUS ADDRESS FOR RESULT  
MOV #TSTCON,ASFR ;ENABLE TEST CONTROL  
BIS #BIT6BIT0,ACSR ;ENABLE DEVICE  
EXIT,REGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.  
;*****  
;RR IF CLEARED  
;*****  
;UNEXPECTED HIGH 14 BITS OF OFFSET REGISTER BIT SET  
;*****  
;RR IF SAME  
;*****
```


TST4	000640R	298	305#																				
TST5	000760R	322	379#																				
TST6	001062R	351#																					
TST7	001136R	359	367#																				
VECTA3	000276R	229#	632*	570*																			
VECTA1	000300R	229#	633*	670	671*																		
VECTB0	000302R	230#	680*	713*																			
VECTB1	000304R	231#	249	698*	713	714*																	
VECTOR	000010R	152#	246																				
WASADR	000104R	186#																					
WBUFEA	000136R	211#																					
WBUFPA	000138R	202#																					
WBUFPO	000140R	202#																					
WBUFSZ	000142R	203#																					
WCR	000256R	220#	260	388*	399	454*	467	572*	601	634*	663	701*											
WDR	000116R	193#																					
WDTD	000114R	192#																					
XFLAG	000005R	150#																					
STEMP	000306R	232#	271*	272	274	284*	285	332*	334	336	345*	346	576*	582*									
STX	= 000021	209#	252#	262#	287#	298	302#	322	326#	348#	359	364#	374	379#									
.	= 005502R	401#	423#	445#	495#	506	511#	545	549#	556	604	609#	616	672#									
.	ARS. 000000 000	679	710#																				
.	005502 001	731#	739#	740#																			

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 XNCBA0,XNCBA0/SQL/CRF:SYM=DDXCOM,XNCBA0
 RUN-TIME: 23.4 SECONDS
 RUN-TIME RATIO: 33/5=7
 CORE USED: 7K (13 PAGES)