

.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

PAGE  
1.

ABSTRACT

"NCB" IS AN "TOMODY" THAT EXERCISES ONE NCV-11A INTERFACE. THE NCV-11A INTERFACE DOES MEMORY INCREMENTS VIA NPR UNTIL A WORD OR BYTE REACHES MAXIMUM CAPACITY AND ATTEMPS TO OVERFLOW. AT THIS TIME AN INTERRUPT IS GENERATED AT BR 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: NCB 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, BR1: 6, DEVCNT: N/A

REQUIRED PARAMETERS:

NONE

6. 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 MAINTENCE 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 INTERATION 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 ENVIROMENT  
-----

THE NCA MODULE IS KNOWN TO OPERATE UNDER THIS ENVIROMENT:

#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		KW11L 1 UNIT
	LP11 1 UNIT		
#3	PDP-11/34 CPU WITH 32K		
	RK11-D 1 DRIVE		
	NCV11A 1 UNIT		
	KW11L 1 UNIT		

```

141 .LIST SEQ,RIN
142 .LIST WC,WD,CND
143 .LIST WE
144 .TITLE NCRA DEC/Y11 SYSTEM EXERCISER MODULE
145 .MACRO VERSION 6 23-MAY-78
146 .LIST RIN
*****
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 ;*****
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 ;*****

```

252  
253  
254  
255 000406 104407 000000  
256 000406 104407 000000  
257 000412 104407 000000  
258 000416 005777 177630  
259 000422 005777 177626  
260 000426 005777 177624  
261 000432 005777 177622  
262 000436 005777 177620  
263 000442 005777 177616  
264 000446 005777 177616  
265  
266  
267  
268 000452 104407 000000  
269 000452 104407 000000  
270 000456 104407 000000  
271 000462 012767 004000 177616  
272 000470 016777 177612 177554  
273 000476 017767 177550 177400  
274 000504 016767 177576 177370  
275 000512 052767 000200 177362  
276 000520 026767 177356 177356  
277 000526 001403  
278  
279 000530 104405 000000 000000  
280  
281 000536 104407 000000  
282 000536 104407 000000  
283 000542 016767 177554  
284 000546 022767 177526  
285 000552 022767 000001 177526  
286 000560 001343  
287  
288  
289  
290 000562 104407 000000  
291 000562 104407 000000  
292 000566 104407 000000  
293 000572 012777 007776 177452  
294 000600 012777 004000 177454  
295 000606 012767 000200 177266  
296 000614 017767 177432 177262  
297 000626 026767 177254 177254  
298 000630 001403  
299  
300 000632 104405 000000 000000  
301

```
*****  
;VERIFY THE HIGH NCV11 BUS ADDRESSES RESPOND  
*****  
TST1: *****  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
TST QCSR ;ADDRESS  
TST QOFF ;  
TST QWCP ;THE  
TST QPAR ;NCV11  
TST QSPR ;  
TST QADM ;  
TST QRR1 ;ADDRESSES  
*****  
;FLOAT A 1 ACROSS 10 BITS OF THE COMMAND/STATUS REG.  
*****  
TST2: *****  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #RIT1,STEMP ;LOAD INITIAL REG. VALUE  
MOV STEM,ACSR ;LOAD CSR REG.  
MOV ACSR,ASTAT ;READ CSR  
MOV STEM,ACSR ;LOAD EXPECTED  
RIS #RIT7,ACSR ;CHANGE THE "READY" BIT  
CMP ACSR,ASTAT ;COMPARE THE VALUES  
REQ 2S ;//RR IF SAME  
*****  
RDERS,REGIN,NULL ;UNEXPECTED VALUE IN THE CSR REGISTER  
*****  
TST3: *****  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #RIT6,ACSR ;LOAD CSR REG.  
MOV #CLRALL,ASFP ;GENERATE "CLR ALL L"  
MOV #RIT7,ACSR ;LOAD EXPECTED VALUE  
MOV ACSR,ASTAT ;READ THE CSR REG.  
CMP ACSR,ASTAT ;COMPARE VALUES  
REQ 1S ;//RR IF NOT  
*****  
;VERIFY THAT "CLEAR ALL" CLEARS THE CSR REGISTER  
*****  
TST3: *****  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #RIT6,ACSR ;LOAD CSR REG.  
MOV #CLRALL,ASFP ;GENERATE "CLR ALL L"  
MOV #RIT7,ACSR ;LOAD EXPECTED VALUE  
MOV ACSR,ASTAT ;READ THE CSR REG.  
CMP ACSR,ASTAT ;COMPARE VALUES  
REQ 1S ;//RR IF SAME  
*****  
RDERS,REGIN,NULL ;"CLR ALL L" FAILED TO CLEAR CSR REG.  
*****
```

302  
303  
304  
305 000640 104407 000000  
306 000640 104407 000000  
307 000644 104407 000000  
308 000650 012777 003636 177374  
309 000656 012767 003600 177216  
310 000664 015077 177362  
311 000670 017767 177356 177206  
312 000676 026767 177200 177200  
313 000704 001403  
314  
315 000706 104405 000000 000000  
316  
317 000714 012777 003636 177330  
318 000722 012767 000236 177152  
319 000730 012767 177332 177142  
320 000734 017767 177332 177134  
321 000742 026767 177134 177134  
322 000750 001403  
323  
324 000752 104405 000000 000000  
325  
326  
327  
328  
329 000760 104407 000000  
330 000764 104407 000000  
331 000770 012767 000020 177310  
332  
333  
334 000776 016777 177304 177256  
335 001004 017767 177252 177072  
336 001012 016767 177270 177062  
337 001020 026767 177056 177056  
338 001026 001403  
339  
340 001030 104405 000000 000000  
341  
342 001036 104407 000000  
343 001040 104407 000000  
344 001046 006267 177234  
345 001052 022767 000001 177226  
346 001060 001346

```
*****  
;VERIFY LOW BYTE OPERATION OF THE "CSR" REGISTER  
*****  
TST4: *****  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #3636,ACSR ;LOAD CSR REGISTER  
MOV #3600,ACSR ;LOAD EXPECTED VALUE  
CLR ACSR ;CLEAR LOW BYTE  
MOV ACSR,ASTAT ;READ STATUS REG.  
CMP ACSR,ASTAT ;COMPARE VALUES  
REQ 1S ;//RR IF SAME  
*****  
RDERS,REGIN,NULL ;CLEARING LOW BYTE OF THE CSR CHANGED THE HIGH BYTE  
*****  
TST5: *****  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #RIT4,STEMP ;LOAD INITIAL REG. VALUE  
*****  
1S: MOV STEM,ACSR ;LOAD SFR REG.  
MOV ACSR,ASTAT ;READ SFR  
MOV STEM,ACSR ;LOAD EXPECTED  
CMP ACSR,ASTAT ;COMPARE THE VALUES  
REQ 2S ;//RR IF SAME  
*****  
RDERS,REGIN,NULL ;UNEXPECTED VALUE IN THE SFR REGISTER  
*****  
2S: *****  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #RIT4,STEMP ;LOAD INITIAL REG. VALUE  
MOV ACSR,ASTAT ;LOAD SFR  
MOV STEM,ACSR ;LOAD EXPECTED  
CMP ACSR,ASTAT ;COMPARE THE VALUES  
REQ 1S ;//RR IF NOT
```

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

0011262 104407 000000  
0011263 104407 000000  
0011272 012777 000014 177152  
0011000 012767 000014 176774  
0011264 012777 000014 177152  
0011120 017767 177144 176764  
0011276 001403  
0011130 104405 000000 000000  
0011136 104407 000000  
0011142 104407 000000  
0011446 012777 000016 177106  
0011454 012777 000016 177106  
0011622 005267 176714  
0011626 017767 177070 176710  
0011174 001403  
0011176 104405 000000 000000  
0012040 104407 000000  
0012100 104407 000000  
0012140 012767 000001 176660  
0012220 012777 004000 177032  
0012300 016777 176646 177026  
0012316 017767 177014 176640  
0012440 006767 176432  
0012520 001403  
0012540 104405 000000 000000  
0012620 104407 000000  
0012660 104407 000000  
0012720 006367 176604  
0012760 001351

```
*****  
;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,ASFR ;LOAD THE S.F. REGISTER  
MOV #14,ACSP ;LOAD EXPECTED VALUE  
CLR #ASFR ;CLEAR HIGH BYTE OF S.F. REG.  
MOV #ASFR,ASTAT ;READ THE REGISTER  
CMP ACSR,ASTAT ;COMPARE THE VALUES  
REQ TS17 ;/RR IF SAME  
*****  
;CLEARING HIGH BYTE OF CSR REG. CHANGED THE LOW BYTE AL  
*****  
;VERIFY THAT "CLEAR ALL" CLEARS THE SFR REGISTER  
*****  
TST7:  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #16,ASFR ;LOAD SFR REG.  
BTS #CLRALL,ASFR ;GENERATE "CLR ALL L"  
CLR ACSR ;LOAD EXPECTED VALUE  
MOV #ASFR,ASTAT ;READ THE SFR REG.  
REQ TS10 ;/RR IF SAME  
*****  
;CLEAR ALL L" FAILED TO CLEAR SFR REG.  
*****  
;FLOAT A 1 ACROSS THE WORD COUNT REGISTER  
*****  
TST10:  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #R10,ACSP ;LOAD EXPECTED VALUE  
15: MOV #CLRALL,ASFR ;RESET THE DEVICE  
MOV ACSR,#WCR ;LOAD WORD COUNT "A"  
MOV #WCR,ASTAT ;READ WORD COUNT  
CMP ACSR,ASTAT ;COMPARE VALUES  
REQ TS ;/RR IF SAME  
*****  
;WORD COUNT REG. IN ERROR  
*****  
25: 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

0013000 104407 000000  
0013004 104407 000000  
0013100 012767 000001 176564  
0013116 012777 004000 176736  
0013124 012777 176552 176722  
0013320 017767 176722 176544  
0013340 026767 176536 176536  
0013440 001403  
0013500 104405 000000 000000  
0013560 104407 000000  
0013620 104407 000000  
0013660 006367 176510  
0013720 001351  
0013740 104407 000000  
0014000 104407 000000  
0014040 012767 000001 176470  
0014120 012777 004000 176642  
0014200 016777 176456 176626  
0014260 017767 176522 176450  
0014340 026767 176442 176442  
0014420 001403  
0014440 104405 000000 000000  
0014520 104407 000000  
0014560 104407 000000  
0014620 006367 176414  
0014660 001351

```
*****  
;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,ASFR ;RESET THE DEVICE  
MOV ACSR,#BAR ;LOAD BUS ADDRESS "A"  
MOV #BAR,ASTAT ;READ BUS ADDRESS  
CMP ACSR,ASTAT ;COMPARE VALUES  
REQ TS ;/RR IF SAME  
*****  
;BUS ADDRESS REG. IN ERROR  
*****  
25: 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  
*****  
;FLOAT A 1 ACROSS THE OFFSET REGISTER  
*****  
TST12:  
BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...  
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV #R10,ACSP ;LOAD EXPECTED VALUE  
15: MOV #CLRALL,ASFR ;RESET THE DEVICE  
MOV ACSR,#OFF ;LOAD OFFSET "A"  
MOV #OFF,ASTAT ;READ OFFSET  
CMP ACSR,ASTAT ;COMPARE VALUES  
REQ TS ;/RR IF SAME  
*****  
;OFFSET REG. IN ERROR  
*****  
25: 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
```







```

XNCBA0-P11 15-NOV-78 11:40
661 003066 104405 000000 000000
662
663 003074 017767 175156 175002 1S:
664 003102 012767 000000 174772
665 003110 026767 174766 174766
666 003116 001403
667
668 003120 104405 000000 000000
669
670 003126 016777 175146 175142 7S:
671 003134 005077 175140
        HDRERS,REGIN,NULL          ;INCORRECT BUS ADDRESS VALUE AFTER 1 512 WORD TRANSFER
        ;*****
        MOV    @WCR,ASTAT        ;READ W.C. REGISTER
        MOV    @C,ACSR          ;LOAD EXPECTED W.C. VALUE
        CMP    @CSR,ASTAT        ;COMPARE VALUES
        BEQ    7S                ;//RR IF SAME
        ;*****
        HDRERS,REGIN,NULL          ;INCORRECT WORD COUNT REGISTER VALUE AFTER 1 512 WORD T
        ;*****
        MOV    @VECTA1,@VECTA0
        CLR    @VECTA1           ;RESET VECTOR
    
```

```

XNCBA0-P11 15-NOV-78 11:40
672
673
674
675 003140
676 003140 104407 000000
677 003144 104407 000000
678 003150 032767 000001 174640
679 003156 001110
680 003160 012700 003470
681 003164 012720 170300
682 003170 104407 000000
683 003174 104407 000000
684 003200 022700 005500
685 003204 001367
686
687 003206 012777 004000 175046
688 003214 104415 000000 000124
689 003222 012777 003336 175052
690 003230 116777 174556 175046
691 003236 016700 174664
692 003242 012700 000003
693 003246 016701 174656
694 003252 006201
695 003254 006201
696 003258 006201
697 003260 006201
698 003262 042701 177774
699 003266 060100
700 003270 010077 174760
701 003274 012777 177766 174754
702 003302 012777 000000 174750
703 003310 012777 000062 174734
704 003316 052777 000002 174736
705 003324 052777 000001 174720
706 003332 104400 000000
707
708 003336 052777 000400 174716 10S:
709 003344 052777 004000 174710
710
711 003352 000004 000000 003360
712
713 003360 016777 174720 174714 11S:
714 003366 005077 174712
715 003372 000240
716 003374 000240
717 003376 000240
        ;*****
        ;MATRIX MODE DATA COLLECTOR
        ;*****
        TST20:
        BREAKS,REGIN            ;TEMPORARY RETURN TO MONITOR...
        BREAKS,REGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
        BIT    #R10,SRI          ;TEST IF INHIBIT IS SET
        BNE    TST21            ;//RR IF SET
        MOV    #RUFPA,R0        ;LOAD BUFFER POINTER
        MOV    #170300,(R0)+    ;CLEAR THE BUFFER
        BREAKS,REGIN            ;TEMPORARY RETURN TO MONITOR...
        BREAKS,REGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
        CMP    #RUF2,R0        ;FINISHED ?
        BNE    1S
        MOV    #CLRALL,@SFR     ;CLEAR THE DEVICE
        GETPAS,REGIN,R0RUFPA    ;GET PHYSICAL ADDRESS FROM 16-BIT R0RUFPA
        MOV    #10,@WCR         ;LOAD INTR. VECTOR
        MOVR   #R1,@VECTB0     ;LOAD BR LEVEL
        MOV    #R1,@VECTB1     ;LOAD BR LEVEL
        MOV    #RUFPA,R0        ;GET BUFFER POINTER
        BIT    #3,R0            ;CLEAR EXT. MEMORY BITS
        MOV    #RUFPA,R1        ;GET EXTENDED MEMORY BITS
        ASR    R1
        ASR    R1
        ASR    R1
        ASR    R1
        BIT    #17774,R1        ;CLEAR OFF EXCESS
        ADD    R1,R0
        MOV    R0,#OFF         ;LOAD COMBINED ADDRESS
        MOV    #10,@WCR        ;LOAD HIGH 16 BITS OF Z
        MOV    #C,@RAR         ;LOAD LOW 16 BITS OF Z
        MOV    #R175122,@CSR    ;SET RES. AND MODE AND CELL INTR. ENABLE
        MOV    #TESTZ,@SFR     ;SET Z PULSES
        BIS    #R10,@CSR       ;ENABLE THE DEVICE
        EXITS,BEGIN            ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
        ;
        BIS    #ENDDMA,@SFR     ;STOP TRANSFERS
        BIS    #CLRALL,@SFR     ;CLEAR NCV11
        ;-----
        PIRQS,REGIN,11S        ;QUEUE UP TO CONTINUE AT 11S AND RTI
        ;-----
        MOV    @VECTB1,@VECTB0 ;RESET THE VECTOR
        CLR    @VECTB1
        NOP
        NOP
        NOP
    
```





TST4	000640R	298	305#																
TST5	000760R	325	329#																
TST6	001062R	351#																	
TST7	001136R	359	367#																
VECTA9	000976R	239#	632*	670*															
VECTA1	000300R	429#	633*	670	671*														
VECTB0	000302R	230#	680*	713*															
VECTB1	000304R	231#	242	690*	713	714*													
VECTOR	000010R	152#	246																
WASADR	000104R	186#																	
WBUFEA	000136R	201#																	
WBUFPA	000134R	202#																	
WBUFRO	000140R	202#																	
WBUFSZ	000142R	203#																	
WCR	000256R	229#	260	388*	399	454*	467	572*	601	634*	663	701*							
WDR	000116R	163#																	
WDT0	000114R	192#																	
XPLAG	000005R	153#																	
STEMP	000306R	422#	271*	272	274	284*	285	332*	334	336	345*	346	576*	592*					
STX	= 000021	209#	252#	266#	287#	298	302#	327	326#	348#	359	364#	374	379#					
		401#	423#	445#	495#	506	511#	545	549#	556	604	609#	616	672#					
		679	719#																
		= 005502R	731#	739#	740#														

. ABS. 000000 000  
 005502 001

ERRORS DETECTED: 1  
 DEFAULT GLOBALS GENERATED: 0  
 XNCBA0,XNCBA0/SQL/CRF:SYM=DDXCOM,XNCBA0  
 RUN-TIME: 23.4 SECONDS  
 RUN-TIME RATIO: 33/5=5.7  
 CORE USED: 7K (13 PAGES)