

NCBA DEC/X11 SYSTEM EXERCISER MODULE
XNCBAO.P11 15-NOV-78 11:40

MACY11 30A(1052) 22-NOV-78 11:22 PAGE 2

.REM _

IDENTIFICATION

PRODUCT CODE: AC-F234A-MC
PRODUCT NAME: CXNCBAO 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 DIGITAL'S
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

NCBA DEC/X11 SYSTEM EXERCISER MODULE MACV11 30A(1052) 22-NOV-78 11:22 PAGE 3
XNCBA0.P11 15-NOV-78 11:40

PAGE
1.

ABSTRACT

"NCB" IS AN "IOMODY" 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: 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, BR1: 6, DEVcnt: N/A

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

NONE.

NCBA DEC/X11 SYSTEM EXERCISER MODULE MACV11 30A(1052) 22-NOV-78 11:22 PAGE 4
XNCBA0.P11 15-NOV-78 11:40

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

SR1 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.

SR1 BIT0 = 1 INHIBIT MATRIX WORD INCREMENT MODE.
SR1 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 TC11 2 DRIVES TM11 1 DRIVE VSV01 1 SCOPE NCV11A 1 UNIT LP11 1 UNIT | #2 | PDP-11/40 CPU WITH 28K RK11-D 1 DRIVE TM11 1 DRIVE NCV11A 1 UNIT KW11L 1 UNIT |
| #3 | PDP-11/34 CPU WITH 32K RK11-D 1 DRIVE NCV11A 1 UNIT KW11L 1 UNIT | | |

```

145      LIST  SEQ,RIN
146      LIST  MC,CND
147      LIST  NE
148      TITLE NCRA DEC/Y11 SYSTEM EXERCISER MODULE
149      PDCOM VERSION 6 21-NOV-78
150      LIST  RIN
*****+
151      D$CIN: 15000000 141516 146512 144
152      XFLG:  ASCII /NCRA / ;MODULE NAME
153      ADDR: 172760+0 ;USED TO KEEP TRACK OF WBUFF USAGE
154      VECTOR: 376+0 ;1ST DEVICE ADDR.
155      R91:  BYTE PRTV6+0 ;1ST DEVICE VECTOR.
156      R92:  BYTE PRTV0+0 ;2ND RR LEVEL.
157      DVIN1: C+1 ;2ND RR LEVEL.
158      S91:  OPEN ;DEVICE INDICATOR 1.
159      S92:  OPEN ;SWITCH REGISTER 1
160      S93:  OPEN ;SWITCH REGISTER 2
161      S94:  OPEN ;SWITCH REGISTER 3
162      S95:  OPEN ;SWITCH REGISTER 4
*****+
163      ICOUNT: 150000 ;STAT'S WORD.
164      INIT: START ;MODULE START ADDR.
165      SPINT: MODSP ;MODULE STACK POINTER.
166      PASCNT: 000000 ;PASS COUNTER.
167      ICOUNT: 00 ;# OF ITERATIONS PER PASS=80.
168      ICOUNT: 00 ;LOC TO COUNT ITERATIONS
169      SOFTNT: 00 ;LOC TO SAVE TOTAL SOFT ERRORS
170      HSOFTNT: 00 ;LOC TO SAVE TOTAL HARD ERRORS
171      SHDPESS: 00 ;LOC TO SAVE SOFT ERRORS PER PASS
172      SHDPASS: 00 ;LOC TO SAVE HARD ERRORS PER PASS
173      SVSNM: 000000 ;# OF SYS ERRORS ACCUMULATED
174      RANDNM: 000000 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
175      RES1: X ;RESERVED FOR MONITOR USE
176      RES2: X ;RESERVED FOR MONITOR USE
177      SVR1: OPEN ;LOC TO SAVE R0.
178      SVR2: OPEN ;LOC TO SAVE R1.
179      SVR3: OPEN ;LOC TO SAVE R2.
180      SVR4: OPEN ;LOC TO SAVE R3.
181      SVR5: OPEN ;LOC TO SAVE R4.
182      SVR6: OPEN ;LOC TO SAVE R5.
183      SVR7: OPEN ;LOC TO SAVE R6.
184      SVR8: OPEN ;LOC TO SAVE R7.
185      SRADP: 000000 ;ADDR OF CURRENT CSR.
186      ACSP:  OPEN ;ADDR OF GOOD DATA, OR
187      WASADP: 000000 ;CONTENTS OF CSR.
188      ASTAT:  OPEN ;ADDR OF BAD DATA, OR
189      ERRTYP: 000000 ;STATUS REG CONTENTS.
190      ASR1:  OPEN ;TYPE OF ERROR.
191      ASR2:  OPEN ;EXPECTED DATA.
192      ASR3:  OPEN ;FACTUAL DATA.
193      ASR4:  OPEN ;RESTART ADDRESS AFTER END OF PASS
194      #STD:  OPEN ;WORDS FROM MEMORY PER ITERATION
195      WOFR:  OPEN ;WORDS FROM MEMORY PER ITERATION
196      INTD:  OPEN ;#OF INTERRUPTS PER ITERATION
197      IDNUM: 00 ;MODULE IDENTIFICATION NUMBER=0

```

```

198      RBUFEVA: BUFO ;READ BUFFER VIRTUAL ADDRESS
199      RBUFPDA: OPEN ;READ BUFFER PHYSICAL ADDRESS
200      RBUFEA: OPEN ;READ BUFFER EA BITS
201      RBUFSZ: 256 ;SIZE OF THE READ BUFFER
202      WRBUFPDA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
203      WRBUFEA: OPEN ;WRITE BUFFER EA BITS
204      WRBUFSZ: 00 ;WRITE BUFFER SIZE REQUESTED
205      CSERCT: 00 ;WRITE BUFFER SIZE AVAILABLE
206      CDWACT: 00 ;#DATA/DATCK ERROR COUNT
207      CSRDCT: 00 ;CDATA/DATCK WORD COUNT
208      CSRDCT: OPEN ;RESERVED FOR FUTURE USE
209      VNDSP: ;*****
210      STYFC: ;*****
211      ARASE=0 ;ARASE=0
212      AVECTI=0 ;AVECTI=0
213      TSTDMA=BIT1 ;TSTDMA=BIT1
214      CSRMEMBITS3 ;CSRMEMBITS3
215      TESZ2=BIT1 ;TESZ2=BIT1
216      TSTCON=BIT2 ;TSTCON=BIT2
217      TSTDMA=BIT3 ;TSTDMA=BIT3
218      CLFALL=BIT11 ;CLFALL=BIT11
219      CSR: APASE ;CSR: APASE
220      OFF: APASE+2 ;OFF: APASE+2
221      WCR: APASE+4 ;WCR: APASE+4
222      PAR: APASE+6 ;PAR: APASE+6
223      SFR: APASE+10 ;SFR: APASE+10
224      AMI: APASE+12 ;AMI: APASE+12
225      JDT: APASE+14 ;JDT: APASE+14
226      RAP1: APASE+16 ;RAP1: APASE+16
227      CSRDB: APASE+1 ;CSRDB: APASE+1
228      SF4HR: ARASE+11 ;SF4HR: ARASE+11
229      VECTAO: AVECT1 ;VECTAO: AVECT1
230      VECTAI: AVECT1+2 ;VECTAI: AVECT1+2
231      VECTRI: AVECT1+4 ;VECTRI: AVECT1+4
232      STEUP: C ;STEUP: C
233      START: C ;START: C
234      RESETPT: ;RESETPT:
235      LOGIC: MOV    #CLPALL,BSFR ;CLEAR NCV11
236      LOGIC: MOV    #CSRP,RP1 ;LOAD ADDRESS POINTER
237      LOGIC: ADDP,RP1 ;SET RS ADDRESS
238      LOGIC: MOV    R1,(R0)+ ;LOAD DEVICE ADDRESS
239      LOGIC: ADD  #2,RP1 ;UPDATE VALUE
240      LOGIC: CMP   RP1,#CSR4R ;TEST IF DONE
241      LOGIC: ENL   1$ ;ENL 1$
242      LOGIC: MOV    ADDR,(R0) ;LOAD BYTE ADDRESSES
243      LOGIC: INC   (R0)+ ;INC (R0)+
244      LOGIC: MOV    ADDR,(R0) ;LOAD BYTE OF SFR
245      LOGIC: ADD  #1,RP1 ;ADD #1,RP1
246      LOGIC: MOV    R1,(R0)+ ;LOAD VECTORS
247      LOGIC: ADD  #1,RP1 ;UPDATE
248      LOGIC: CMP   #VECTR1+2,R0 ;RR UNTIL DONE
249      LOGIC: MOV    2$ ;LOAD FOR TYPE OUT
250      LOGIC: ADDP,CSRA ;LOAD FOR TYPE OUT
251      LOGIC: MOV    ADDR,CSRA

```

NCBA DEC/X11 SYSTEM EXERCISER MODULE
XNCBAO.P11 15-NOV-78 11:40

MACV11 30A(1052) 22-NOV-78 11:22 PAGE 7

2008

```

*****VERIFY THE H NCV11 BUS ADDRESSES RESPOND*****
*****TST1: *****

000406* 104407 000000*          BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
000412* 104407 000000*          BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
000416* 005777 177630          TST    ACSPR ;ADDRESS
000422* 005777 177626          TST    OFF
000426* 005777 177624          TST    ACWP ;THE
000432* 005777 177622          TST    AWPB
000436* 005777 177620          TST    ACSPR ;NCV11
000442* 005777 177616          TST    AWPB
000446* 005777 177616          TST    ACWP1 ;ADDRESSES
000452*          177616          TST?   ;TEMPORARY RETURN TO MONITOR...
000456* 104407 000000*          BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
000462* 012767 004090 177616      MOV    #H11,STEMP ;LOAD INITIAL REG. VALUE
000470* 016776 177620          MOV    STEMPS,ACSPR ;READ CSC REG.
000474* 017767 177554 1S:       MOV    STEMPS,ASTAT ;READ EXPECTED
000478* 017767 177554          MOV    STEMPS,ACSPR ;LOAD EXPECTED
000482* 017767 177554          MOV    STEMPS,ASTAT ;COMPARE THE "READY" BIT
000486* 017767 177554          PIS    B17,ACSPR ;COMPARE THE VALUES
000490* 017767 177554          CMP    ACSPR,ASTAT ;COMPARE THE VALUES
000494* 017767 177554          REQ    2S ;//BR IF SAME
000500* 017767 177554          RDRS   BEGIN, NULL ;UNEXPECTED VALUE IN THE CSP REGISTER
000504* 001403          2S:      ;UNEXPECTED VALUE IN THE CSP REGISTER
000536* 104407 000000*          BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
000542* 104407 000000*          BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
000546* 006267 177534          ASR    STEMPS ;TEST IF NOW BIT
000550* 006267 177534          CMP    #1,STEMPS ;TEST IF NOT
000554* 006267 177534          RNE    ;BR IF NOT
000560* 001343          ;VERIFY THAT "CLEAR ALL" CLEARS THE CSP REGISTER
000562* 104407 000000*          BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
000566* 104407 000000*          BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
000570* 032777 177452          MOV    #H776,ACSPR ;LOAD A REG.
000574* 032777 177452          MOV    HCLRALL,RSFP ;GENERATE "CLEAR ALL L"
000578* 032777 177452          MOV    #H776,ACSPR ;LOAD EXPECTED VALUE
000582* 032777 177452          MOV    ACSPR,ASTAT ;READ THE CSP REG.
000586* 032777 177452          CMP    ACSPR,ASTAT ;COMPARE VALUES
000590* 001403          ;//BR IF SAME
000632* 104405 000000* 000000* RDRS   BEGIN, NULL ;"CLR ALL L" FAILED TO CLEAR CSP REG.
000636* 104405 000000* 000000*

```

NCBA DEC/X11 SYSTEM EXERCISER MODULE
XNCBAD.P11 15-NOV-78 11:40

MACY11 30A(1052) 22-NOV-78 11:22 PAGE 8

850.0037

```

*****VERIFY LOW BYTE OPERATION OF THE "CSR" REGISTER*****
*****TST4:*****  

000640- 000640 104407 000000- BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR  

000644- 000644 104407 000030- BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

000650- 012777 003636 177374 MOV #3636,ACSR ;LOAD CSR REGISTER  

000656- 012767 003600 177216 MOV #3600,ACSR ;LOAD EXPECTED VALUE  

000664- 010577 177362 CLRR ACSR ;CLEAR LOW BYTE  

000670- 0117767 177356 177206 MOV ACSR,ASTAT ;READ STATUS REG.  

000676- 026767 177200 177200 CMP ACSR,ASTAT ;COMPARE VALUES  

000704- 001403 BEQ 15 ;JUMP IF SAME  

*****TST5:*****  

000706- 104405 000000- 000000 HRDRS,REGIN,NULL ;CLEARING LOW BYTE OF THE CSR CHANGED THE HIGH BYTE  

000714- 012777 003636 177330 15: MOV #3636,ACSR ;LOAD CSR REGISTER  

000722- 012767 000236 B236,ACSR ;LOAD EXPECTED VALUE  

000730- 010577 177336 CLRR ACSRRA ;CLEAR HIGH BYTE OF THE CSR  

000734- 0117767 177312 177142 MOV ACSR,ASTAT ;READ STATUS  

000742- 026767 177134 177134 CMP ACSR,ASTAT ;COMPARE VALUES  

000750- 001403 BEQ 15 ;JUMP IF SAME  

*****TST6:*****  

000752- 104405 000000- 000000 HRDRS,REGIN,NULL ;CLEARING HIGH BYTE OF CSR CHANGED THE LOW BYTE  

*****TST7:*****  

000760- 104407 000000- 000000 ;FLOAT A 1 ACROSS 4 BITS OF THE SPECIAL FUNCTION REGISTER  

000764- 104404 000000- 000020 TST5:  

000770- 012767 000200 177310 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR  

000776- 011777 177304 177256 15: MOV #H14,STEMP ;THEN CONTINUE AT NEXT INSTRUCTION.  

000782- 012767 000200 177310 MOV STEMPSR ;LOAD INITIAL REG. VALUE  

000786- 011777 177252 177172 MOV ASFR,ASTAT ;LOAD SFR PEG.  

001012- 011767 177270 177062 MOV STEMPSR ;READ SFR  

001019- 026767 177056 177056 CMP ACSR,ASTAT ;LOAD EXPECTED  

001026- 001403 REQ 25 ;COMPARE THE VALUES  

001030- 104405 000000- 000000 *****TST8:*****  

001036- 104407 000000- 000000 HRDRS,REGIN,NULL ;UNEXPECTED VALUE IN THE SFR REGISTER  

001042- 104407 000000- 000000 ;  

001045- 002627 177234 177226 25:  

001052- 022767 000001 177226 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR  

001060- 001346 000000- 000000 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

001066- 002627 177234 177226 ASR STEMPSR ;TRY THE NEXT DATA BIT  

001073- 001346 000000- 000000 CMP #1,STEMP ;TEST IF NOW BIT 0  

001080- 001346 000000- 000000 BNE 15 ;BR IF NOT

```

```

344 ;*****  

345 ;VERIFY THAT CLEARING HIGH BYTE OF SFR DOES NOT CLEAR LOW BYTE  

346 ;*****  

347 TST6: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

348 MOV #14,ACSPR ;THEN CONTINUE AT NEXT INSTRUCTION.  

349 WCR #14,ACSPR ;LOAD THE S.F. REGISTER.  

350 CLR @SPR,ASTAT ;LOAD THE EXPECTED VALUE.  

351 MOV @SPR,ASTAT ;CLEAR HIGH BYTE OF S.F. REG.  

352 CMP ACSPR,ASTAT ;READ THE REGISTER.  

353 BEQ TST7 ;COMPARE THE VALUES  

354 TST7: //BR IF SAME  

355 HRDERS,BEGIN,NULL ;CLEARING HIGH BYTE OF CSR REG. CHANGED THE LOW BYTE AL  

356 ;*****  

357 ;VERIFY THAT "CLEAR ALL" CLEARS THE SPR REGISTER  

358 ;*****  

359 TST7: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

360 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

361 MOV #16,ACSPR ;LOAD SPR REG.  

362 BIS #FC,BALL,ACSPR ;GENERATE "CLR ALL L"  

363 CLR ACSPR ;LOAD EXPECTED VALUE.  

364 MOV @SPR,ASTAT ;READ THE SPR REG.  

365 BEQ TST16 ;//BR IF SAME  

366 HRDERS,BEGIN,NULL ;"CLR ALL L" FAILED TO CLEAR SPR REG.  

367 ;*****  

368 ;FLOAT A 1 ACROSS THE WORD COUNT REGISTER  

369 ;*****  

370 TST10: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

371 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

372 MOV #B10,ACSPR ;LOAD EXPECTED VALUE.  

373 001136* 104407 000000* 000000*  

374 001145* 012777 000015 177106  

375 001154* 012777 004212 177106  

376 001163* 035067 176714 176716  

377 001174* 001403 177070 176716  

378 001176* 104405 000000* 000000*  

379 ;*****  

380 ;FLOAT A 1 ACROSS THE WORD COUNT REGISTER  

381 ;*****  

382 TST10: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

383 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

384 MOV #B10,ACSPR ;LOAD EXPECTED VALUE.  

385 001204* 104407 000000* 000000*  

386 001214* 012767 000001 176660  

387 001222* 012777 004000 177332  

388 001230* 016777 176646 177320  

389 001238* 012772 172214 176640  

390 001249* 032767 176432 176632  

391 001252* 001403 177070 176632  

392 001254* 104405 000000* 000000*  

393 ;*****  

394 HRDERS,BEGIN,NULL ;WORD COUNT REG. IN ERROR  

395 ;*****  

396 001252*  

397 001262* 104407 000000*  

398 001266* 104407 000000*  

399 001272* 006367 176604  

400 001276* 001351  

401 ;*****  

402 ;FLOAT A 1 ACROSS THE WORD COUNT REGISTER  

403 ;*****  

404 TST11: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

405 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

406 MOV #B10,ACSPR ;LOAD EXPECTED VALUE.  

407 001300* 104407 000000* 000000*  

408 001304* 012767 000001 176564  

409 001310* 012767 000001 176564  

410 001316* 012777 004000 176736  

411 001324* 012777 176553 176726  

412 001332* 012777 176722 176544  

413 001340* 032767 176536 176534  

414 001348* 001403 177070 176534  

415 001350* 104405 000000* 000000*  

416 ;*****  

417 HRDERS,BEGIN,NULL ;BUS ADDRESS REG. IN ERROR  

418 ;*****  

419 001356*  

420 001323* 104407 000000*  

421 001366* 006367 176510  

422 001372* 001351  

423 ;*****  

424 ;FLOAT A 1 ACROSS THE OFFSET REGISTER  

425 ;*****  

426 TST12: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

427 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

428 MOV #B10,ACSPR ;LOAD EXPECTED VALUE.  

429 001404* 012767 000001 176470  

430 001412* 012777 004000 176542  

431 001420* 012777 176456 176624  

432 001426* 012777 176422 176450  

433 001434* 032767 176442 176442  

434 001442* 001403 177070 176442  

435 001444* 104405 000000* 000000*  

436 ;*****  

437 HRDERS,BEGIN,NULL ;OFFSET REG. IN ERROR  

438 ;*****  

439 001452*  

440 001456* 104407 000000*  

441 001462* 006367 176414  

442 001466* 001351  

443 ;*****  

444 001466*  

445 ;*****  

446 ;FLOAT A 1 ACROSS THE WORD COUNT REGISTER  

447 ;*****  

448 TST12: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

449 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

450 ASL ACSPR ;CHANGE THE DATA.  

451 RNE 1S ;BR IF MORE DATA TO LOAD

```

```

446 ;*****  

447 ;FLOAT A 1 ACROSS THE WORD COUNT REGISTER  

448 ;*****  

449 TST11: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

450 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

451 MOV #B10,ACSPR ;LOAD EXPECTED VALUE.  

452 001300* 104407 000000* 000000*  

453 001304* 012767 000001 176564  

454 001310* 012767 000001 176564  

455 001316* 012777 004000 176736  

456 001324* 012777 176553 176726  

457 001332* 012777 176722 176544  

458 001340* 032767 176536 176534  

459 001348* 001403 177070 176534  

460 001350* 104405 000000* 000000*  

461 ;*****  

462 HRDERS,BEGIN,NULL ;BUS ADDRESS REG. IN ERROR  

463 ;*****  

464 001356*  

465 001323* 104407 000000*  

466 001366* 006367 176510  

467 001372* 001351  

468 ;*****  

469 ;FLOAT A 1 ACROSS THE WORD COUNT REGISTER  

470 ;*****  

471 TST12: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

472 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

473 MOV #B10,ACSPR ;LOAD EXPECTED VALUE.  

474 001404* 012767 000001 176470  

475 001412* 012777 004000 176542  

476 001420* 012777 176456 176624  

477 001426* 012777 176422 176450  

478 001434* 032767 176442 176442  

479 001442* 001403 177070 176442  

480 001444* 104405 000000* 000000*  

481 ;*****  

482 HRDERS,BEGIN,NULL ;OFFSET REG. IN ERROR  

483 ;*****  

484 001452*  

485 001456* 104407 000000*  

486 001462* 006367 176414  

487 001466* 001351  

488 ;*****  

489 001466*  

490 ;*****  

491 ;FLOAT A 1 ACROSS THE WORD COUNT REGISTER  

492 ;*****  

493 TST12: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.  

494 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

495 ASL ACSPR ;CHANGE THE DATA.  

496 RNE 1S ;BR IF MORE DATA TO LOAD

```

```

445
446
447
448 001470* 104407 000000- ;VERIFY NO DUAL REGISTER SELECTION
449 001470* 104407 000000-
450 FST13: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
451           BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
452           MOV #CLPALL,@SPR ;CLEAR THE DEVICE
453           MOV #111111,0OFF ;LOAD OFFSET REGISTER
454           MOV #233332,0WCR ;LOAD W.C. REGISTER
455           MOV #333333,0CSR ;LOAD SPCIAL FUNCTION REGISTER
456           MOV #7636,ACSR ;LOAD COMMAND/STATUS REGISTER
457           ;NOW READ EACH REGISTER AND CHECK THE VALUE
458           MOV #111111,ACSP ;LOAD EXPECTED VALUE
459           MOV #0OFF,ASTAT ;READ A OFFSET REG.
460           CMP ACSP,ASTAT ;COMPARE
461           REQ 15 ;//BR IF SAME
462           001403
463 FST14: HRSERS,BEGIN,NULL ;DUAL ADDRESS ERROR
464 001570* 104405 000000- 000000
465           MOV #222222,ACSR ;LOAD EXPECTED VALUE
466           MOV #0B6F,ASTAT ;READ B.A. REG.
467           CMP ACSP,ASTAT ;COMPARE
468           026767 176264 176264
469           001403
470 FST15: HRSERS,BEGIN,NULL ;DUAL ADDRESS ERROR
471 001622* 104405 000000- 000000
472           MOV #333333,ACSP ;LOAD EXPECTED VALUE
473 001630* 012767 033333 176244 25:
474           MOV #176416,ACSP ;LOAD EXPECTED VALUE
475           CMP ACSP,ASTAT ;COMPARE
476           026767 176232 176232
477           001403
478 001654* 104405 000000- 000000
479           HRSERS,BEGIN,NULL ;DUAL ADDRESS ERROR
480           MOV #36,ACSP ;LOAD EXPECTED VALUE
481 001662* 012767 000036 176212 35:
482           MOV #0SF6,ASTAT ;READ SPECIAL FUNCTION REG
483           CMP ACSP,ASTAT ;COMPARE
484           001704* 001403
485 FST16: HRSERS,BEGIN,NULL ;DUAL ADDRESS ERROR
486 001714* 012767 007636 176150 45:
487           MOV #7636,ACSR ;LOAD EXPECTED VALUE
488           MOV #ACSP,ASTAT ;READ COMMAND/STATUS REGISTER
489           CMP ACSP,ASTAT ;COMPARE
490           001736* 001403
491 FST17: HRSERS,BEGIN,NULL ;DUAL ADDRESS ERROR
492 001740* 104405 000000- 000000
493           MOV #CLPALL,@SPR ;CLEAR THE DEVICE
494

```

```

495
496
497
498 001754* ;VERIFY "CLR ALL" CLEARS THE EXTENDED OFFSET BITS
499 001754* 104407 000000-
500 001760* 104407 000000-
501 001762* 005067 000000
502 001764* 005067 000000
503 001766* 012777 000003 176264
504 002004* 012777 004000 176250
505 002012* 012767 176236 176064
506 002020* 001403
507 FST18: HRSERS,BEGIN,NULL ;CLEAR ALL FAILED TO CLEAR EXTENDED OFFSET REGISTER
508 002022* 104405 000000- 000000
509
510
511
512
513 FST19: TEST THE "ACTIVE" FLOP CAN SET AND CLEAR
514 002030* ;TEMPORARY RETURN TO MONITOR
515 002030* 104407 000000- ;THEN CONTINUE AT NEXT INSTRUCTION.
516           BREAKS,BEGIN ;CLEAR EXPECTED VALUE
517           MOV #CLPALL,@SPR ;CLEAR THE DEVICE
518           BIS #STDMA,@SPR ;SET THE "TEST DMA" TO PREVENT DATA TRANSFERS
519           BIS #BIT4#BIT1,ACSR ;SET "MATRIX MODE" AND "RTVE" MODE
520           ;NOW SET THE "ACTIVE" FLOP AND VERIFY "INTERFACE IDLE" GOES LOW
521           MOV #BIT4#BIT1,ACSP ;LOAD EXPECTED VALUE
522           BIS #BIT0,ACSR ;SET "ACTIVE" TO A 1
523           MOV #ACSP,ASTAT ;READ STATUS
524           CMP ACSP,ASTAT ;COMPARE VALUES
525           002112* 001406 176140
526           002114* 052777 000400 176140
527           BIS #ENDDMA,@SPR ;STOP DMA IF POSSIBLE
528 FST20: HRSERS,BEGIN,NULL ;"ACTIVE" FLOP FAILED TO SET
529
530 002130* 052777 000400 176124 15: ;POKE THE "END DMA" SIGNAL AND VERIFY "ACTIVE" CLEARS
531           BIS #ENDDMA,@SPR ;SEND "END DMA" SIGNAL
532           MOV #BIT7#BIT4#BIT1,ACSR ;SET "INTERFACE IDLE" INTO EXPECTED
533           MOV #ACSP,ASTAT ;READ STATUS
534           CMP ACSP,ASTAT ;COMPARE
535           BIS 25 ;//RR IF SAME
536 FST21: HRSERS,BEGIN,NULL ;"END DMA" AGAIN FAILED TO CLEAR "ACTIVE" FLOP
537 002162* 104405 000000- 000000
538           BIS #ENDDMA,@SPR ;POKE "ENABLE DMA"
539           ;POKE "ENABLE DMA" AND THEN ISSUE "CLR ALL" SIGNAL TO ENSURE THE "ACTIVE" FLOP CLEARS
540 002170* 012767 000200 175704 25: ;LOAD EXPECTED
541           BIS #BIT0,ACSR ;POKE "ENABLE DMA"
542           BIS #CLPALL,@SPR ;GENERATE "CLR ALL"
543           MOV #ACSP,ASTAT ;READ STATUS
544           CMP ACSP,ASTAT ;COMPARE
545           002226* 001403
546 FST22: HRSERS,BEGIN,NULL ;"CLR ALL" FAILED TO CLEAR "ACTIVE AND PENDING" FLOPS
547 002230* 104405 000000- 000000
548

```

```

549
550
551
552 3322367 104407 000000- 000000
553 0022427 104407 000000- 000002 175542
554 0022427 032767 000002 175542
555 0222547 011146 000000- 000000
556 0222547 011146 034903 175776
557 0222567 011146 034903 175776
558 0222567 011146 034903 175776
559 0222567 011146 034903 175776
560 0222567 011146 034903 175776
561 0222567 011146 034903 175776
562 00223047 011367 000000- 000000
563 00223047 011367 004500- 000000
564 00223127 104415 000000- 000124
565 0022320 316700 175604
566 00223247 006200
567 00223267 006200
568 00223307 006200
569 00223347 006200
570 00223347 006200
571 00223347 0117774 175774
572 00223347 0117774 175774 175774
573 00223347 0117774 175774 175774
574 00223347 0117774 175774 175774
575 00223347 0117774 175774 175774
576 00223347 0117774 175774 175774
577 00223347 0117774 175774 175774
578 00223347 0117774 175774 175774
579 00223347 0117774 175774 175774
580 00223347 0117774 175774 175774
581 00223347 0117774 175774 175774
582 00223347 0117774 175774 175774
583 00223347 0117774 175774 175774
584 00223347 0117774 175774 175774
585 00224427 005667 175434
586 00224467 0117767 175602 175430
587 00224547 042767 000003 175422
588 00224627 001403
589 00224647 104405 000000- 000000
590 00224727 0117777 004000 175562
591 00224727 0117777 004000 175562
592 00224727 0117777 004000 175562
593 00224727 0117777 004000 175562
594 00224727 0117777 004000 175562
595 00224727 0117777 004000 175562
596 00224727 0117777 004000 175562
597 00224727 001403
598 00225327 104405 000000- 000000
599 00225327 104405 000000- 000000
600 00225327 104405 000000- 000000
601 00225327 0117767 175512 175336
602 00225327 0117767 175512 175336
603 00225327 0117767 175512 175336
604 00225327 001403

```

ENABLE A 512 WORD TRANSFER SECTION LIST MODE (MAINT MODE)

TST16:
 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR.
 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
 BIT #BIT1,SR1 ;TEST IF INHIBIT TEST IS SET
 BNE TST17,175774 ;/BR IF SET
 MOV #CLPALL,ASFR ;CLEAR THE DEVICE
 MOV #BUF0,RO ;LOAD BUFFER POINTER
 MOV #7070,(RO)+ ;PRESET THE BUFFER WITH DATA
 45: BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR.
 CMP RO,#BUF1 ;TEST IF DONE
 BNE 45 ;/BR IF NOT
 GETPAS,REGIN,RRUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RRUFVA
 MOV RRUFVA,RO ;GET EXTENDED ADDRESS BITS
 ASR RO ;
 ASR RO ;
 ASR RO ;
 ASR RO ;
 BCC #177774,RO ;CLEAR OFF EXCESSIVE
 MOV #0,OFF ;LOAD EXTENDED BITS
 MOV #-512,ACCR ;SET UP 512 WORD TRANSFER
 MOV RRUPA,ACAR ;LOAD BUS ADDRESS FOR RESULT
 MOV #TSTD,WTSTCON,ASFR ;ENABLE TEST CONTROL AND DMA FLOPS
 BIS #AITO,ACSR ;ENABLE DEVICE
 MOV #512,STMP ;LOAD THE COUNTER
 BIS #TESTZ,ASFR ;ENABLE INT Z
 BIS #TESTZ,ASFR ;DISABLE INT Z
 BIS #AITO,ACSR ;ALLOCATE DMA TRANSFER
 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR.
 DEC #TEMP ;THEN CONTINUE AT NEXT INSTRUCTION.
 DEC #TEMP ;FINISHED ALL WORDS?
 BNE 35 ;/BR UNTIL DONE
 ;THE TRANSFER IS NOW COMPLETE
 CLF ACSP ;CLEAR EXPECTED
 MOV #OFF,ASTAT ;READ OFFSET REG.
 BIC #3,ASTAT ;CLEAR LOW 2 BITS
 BEQ 25 ;/BR IF HIGH 14 BITS ARE CLEARED
 HRDERS,REGIN,NULL ;UNEXPECTED OFFSET REGISTER BIT SET

 TST17:
 BREAKS,REGIN,RRUFVA ;INCORRECT WORD COUNT REGISTER VALUE AFTER A 1 WORD TRA
 HRDERS,REGIN,NULL ;INCORRECT BUS ADDRESS VALUE AFTER A 1 WORD TRANSFER
 MOV #0,ACSP,ASTAT ;READ W.C. REGISTER
 CMP #0,ACSP,ASTAT ;LOAD EXPECTED W.C. VALUE
 BEQ TST17 ;/BR IF SAME

```

605 00225647 104405 000000- 000000
606 00225647 0117767 000000- 000000
607 00225647 0117767 000000- 000000
608 00225647 0117767 000000- 000000
609 00225647 0117767 000000- 000000
610 00225647 0117767 000000- 000000
611 00225647 0117767 000000- 000000
612 00225647 0117767 000000- 000000
613 00225647 0117767 000000- 000000
614 00225647 0117767 000000- 000000
615 00225647 0117767 000000- 000000
616 00225647 0117767 000000- 000000
617 00225647 0117767 000000- 000000
618 00225647 0117767 000000- 000000
619 00225647 0117767 000000- 000000
620 00225647 0117767 000000- 000000
621 00225647 0117767 000000- 000000
622 00225647 0117767 000000- 000000
623 00225647 0117767 000000- 000000
624 00225647 0117767 000000- 000000
625 00225647 0117767 000000- 000124
626 00225647 0117767 000000- 000124
627 00225647 0117767 000000- 000124
628 00225647 0117767 000000- 000124
629 00225647 0117767 000000- 000124
630 00225647 0117767 000000- 000124
631 00225647 0117767 000000- 000124
632 00225647 0117767 000000- 000124
633 00225647 0117767 000000- 000124
634 00225647 0117767 000000- 000124
635 00225647 0117767 000000- 000124
636 00225647 0117767 000000- 000124
637 00225647 0117767 000000- 000124
638 00225647 0117767 000000- 000124
639 00225647 0117767 000000- 000124
640 00225647 0117767 000000- 000124
641 00225647 0117767 000000- 000124
642 00225647 0117767 000000- 000124
643 00225647 0117767 000000- 000124
644 00225647 0117767 000000- 000124
645 003000 000004 000000- 003006*
646 003000 000004 000000- 003006*
647 003000 000004 000000- 003006*
648 003000 000004 000000- 003006*
649 003000 000004 000000- 003006*
650 003000 000004 000000- 003006*
651 003000 000004 000000- 003006*
652 003000 000004 000000- 003006*
653 003000 000004 000000- 003006*
654 003000 000004 000000- 003006*
655 003000 000004 000000- 003006*
656 003000 000004 000000- 003006*
657 003000 000004 000000- 003006*
658 003000 000004 000000- 003006*
659 003000 000004 000000- 003006*
660 003000 000004 000000- 003006*
661 003000 000004 000000- 003006*
662 003000 000004 000000- 003006*
663 003000 000004 000000- 003006*
664 003000 000004 000000- 003006*
665 003000 000004 000000- 003006*
666 003000 000004 000000- 003006*
667 003000 000004 000000- 003006*
668 003000 000004 000000- 003006*
669 003000 000004 000000- 003006*
670 003000 000004 000000- 003006*
671 003000 000004 000000- 003006*
672 003000 000004 000000- 003006*
673 003000 000004 000000- 003006*
674 003000 000004 000000- 003006*
675 003000 000004 000000- 003006*
676 003000 000004 000000- 003006*
677 003000 000004 000000- 003006*
678 003000 000004 000000- 003006*
679 003000 000004 000000- 003006*
680 003000 000004 000000- 003006*
681 003000 000004 000000- 003006*
682 003000 000004 000000- 003006*
683 003000 000004 000000- 003006*
684 003000 000004 000000- 003006*
685 003000 000004 000000- 003006*
686 003000 000004 000000- 003006*
687 003000 000004 000000- 003006*
688 003000 000004 000000- 003006*
689 003000 000004 000000- 003006*
690 003000 000004 000000- 003006*
691 003000 000004 000000- 003006*
692 003000 000004 000000- 003006*
693 003000 000004 000000- 003006*
694 003000 000004 000000- 003006*
695 003000 000004 000000- 003006*
696 003000 000004 000000- 003006*
697 003000 000004 000000- 003006*
698 003000 000004 000000- 003006*
699 003000 000004 000000- 003006*
700 003000 000004 000000- 003006*
701 003000 000004 000000- 003006*
702 003000 000004 000000- 003006*
703 003000 000004 000000- 003006*
704 003000 000004 000000- 003006*
705 003000 000004 000000- 003006*
706 003000 000004 000000- 003006*
707 003000 000004 000000- 003006*
708 003000 000004 000000- 003006*
709 003000 000004 000000- 003006*
710 003000 000004 000000- 003006*
711 003000 000004 000000- 003006*
712 003000 000004 000000- 003006*
713 003000 000004 000000- 003006*
714 003000 000004 000000- 003006*
715 003000 000004 000000- 003006*
716 003000 000004 000000- 003006*
717 003000 000004 000000- 003006*
718 003000 000004 000000- 003006*
719 003000 000004 000000- 003006*
720 003000 000004 000000- 003006*
721 003000 000004 000000- 003006*
722 003000 000004 000000- 003006*
723 003000 000004 000000- 003006*
724 003000 000004 000000- 003006*
725 003000 000004 000000- 003006*
726 003000 000004 000000- 003006*
727 003000 000004 000000- 003006*
728 003000 000004 000000- 003006*
729 003000 000004 000000- 003006*
730 003000 000004 000000- 003006*
731 003000 000004 000000- 003006*
732 003000 000004 000000- 003006*
733 003000 000004 000000- 003006*
734 003000 000004 000000- 003006*
735 003000 000004 000000- 003006*
736 003000 000004 000000- 003006*
737 003000 000004 000000- 003006*
738 003000 000004 000000- 003006*
739 003000 000004 000000- 003006*
740 003000 000004 000000- 003006*
741 003000 000004 000000- 003006*
742 003000 000004 000000- 003006*
743 003000 000004 000000- 003006*
744 003000 000004 000000- 003006*
745 003000 000004 000000- 003006*
746 003000 000004 000000- 003006*
747 003000 000004 000000- 003006*
748 003000 000004 000000- 003006*
749 003000 000004 000000- 003006*
750 003000 000004 000000- 003006*
751 003000 000004 000000- 003006*
752 003000 000004 000000- 003006*
753 003000 000004 000000- 003006*
754 003000 000004 000000- 003006*
755 003000 000004 000000- 003006*
756 003000 000004 000000- 003006*
757 003000 000004 000000- 003006*
758 003000 000004 000000- 003006*
759 003000 000004 000000- 003006*
760 003000 000004 000000- 003006*
761 003000 000004 000000- 003006*
762 003000 000004 000000- 003006*
763 003000 000004 000000- 003006*
764 003000 000004 000000- 003006*
765 003000 000004 000000- 003006*
766 003000 000004 000000- 003006*
767 003000 000004 000000- 003006*
768 003000 000004 000000- 003006*
769 003000 000004 000000- 003006*
770 003000 000004 000000- 003006*
771 003000 000004 000000- 003006*
772 003000 000004 000000- 003006*
773 003000 000004 000000- 003006*
774 003000 000004 000000- 003006*
775 003000 000004 000000- 003006*
776 003000 000004 000000- 003006*
777 003000 000004 000000- 003006*
778 003000 000004 000000- 003006*
779 003000 000004 000000- 003006*
780 003000 000004 000000- 003006*
781 003000 000004 000000- 003006*
782 003000 000004 000000- 003006*
783 003000 000004 000000- 003006*
784 003000 000004 000000- 003006*
785 003000 000004 000000- 003006*
786 003000 000004 000000- 003006*
787 003000 000004 000000- 003006*
788 003000 000004 000000- 003006*
789 003000 000004 000000- 003006*
790 003000 000004 000000- 003006*
791 003000 000004 000000- 003006*
792 003000 000004 000000- 003006*
793 003000 000004 000000- 003006*
794 003000 000004 000000- 003006*
795 003000 000004 000000- 003006*
796 003000 000004 000000- 003006*
797 003000 000004 000000- 003006*
798 003000 000004 000000- 003006*
799 003000 000004 000000- 003006*
800 003000 000004 000000- 003006*
801 003000 000004 000000- 003006*
802 003000 000004 000000- 003006*
803 003000 000004 000000- 003006*
804 003000 000004 000000- 003006*
805 003000 000004 000000- 003006*
806 003000 000004 000000- 003006*
807 003000 000004 000000- 003006*
808 003000 000004 000000- 003006*
809 003000 000004 000000- 003006*
810 003000 000004 000000- 003006*
811 003000 000004 000000- 003006*
812 003000 000004 000000- 003006*
813 003000 000004 000000- 003006*
814 003000 000004 000000- 003006*
815 003000 000004 000000- 003006*
816 003000 000004 000000- 003006*
817 003000 000004 000000- 003006*
818 003000 000004 000000- 003006*
819 003000 000004 000000- 003006*
820 003000 000004 000000- 003006*
821 003000 000004 000000- 003006*
822 003000 000004 000000- 003006*
823 003000 000004 000000- 003006*
824 003000 000004 000000- 003006*
825 003000 000004 000000- 003006*
826 003000 000004 000000- 003006*
827 003000 000004 000000- 003006*
828 003000 000004 000000- 003006*
829 003000 000004 000000- 003006*
830 003000 000004 000000- 003006*
831 003000 000004 000000- 003006*
832 003000 000004 000000- 003006*
833 003000 000004 000000- 003006*
834 003000 000004 000000- 003006*
835 003000 000004 000000- 003006*
836 003000 000004 000000- 003006*
837 003000 000004 000000- 003006*
838 003000 000004 000000- 003006*
839 003000 000004 000000- 003006*
840 003000 000004 000000- 003006*
841 003000 000004 000000- 003006*
842 003000 000004 000000- 003006*
843 003000 000004 000000- 003006*
844 003000 000004 000000- 003006*
845 003000 000004 000000- 003006*
846 003000 000004 000000- 003006*
847 003000 000004 000000- 003006*
848 003000 000004 000000- 003006*
849 003000 000004 000000- 003006*
850 003000 000004 000000- 003006*
851 003000 000004 000000- 003006*
852 003000 000004 000000- 003006*
853 003000 000004 000000- 003006*
854 003000 000004 
```

NCBA DEC/X11 SYSTEM EXERCISER MODULE XNCBA0.P11 15-NOV-78 11:40

MACV11 30A(1052) 22-NOV-78 11:22 PAGE 15

SEQ 0014

```
661 003066* 104405 000000* 000000 HRDERS,REGIN,NULL ;INCORRECT RWS ADDRESS VALUE AFTER A 512 WORD TRANSFER
662 003074* 017767 175156 175002 1S: MOV #WCR,ASTAT ;READ W.C. REGISTER
663 003100* 012767 174766 174768 MOV #ACSR,ASTAT ;LOAD EXPECTED W.C. VALUE
664 003110* 026767 174766 174768 CMP ACSR,ASTAT ;COMPARE VALUES
665 003116* 001403 BEQ 7S ;//RR IF SAME
666 003120* 104405 000000* 000000 HRDERS,REGIN,NULL ;INCORRECT WORD COUNT REGISTER VALUE AFTER A 512 WORD T
667 003125* 016777 175146 175142 7S: MOV VECTBL,VECTRA ;RESET VECTOR
668 003134* 005077 175140 CLR VECTRA ;RESET VECTOR
```

NCBA DEC/X11 SYSTEM EXERCISER MODULE XNCRA0.P11 15-NOV-78 11:40

MACV11 30A(1052) 22-NOV-78 11:22 PAGE 16

SEQ 0015

```
;*****  
;MATRIX MODE DATA COLLECTOR  
;*****  
FST20:  
    BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR.  
    BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
    BNE #RT10,SR1 ;SET IF INITRIT IS SET  
    BNE TST21 ;//BR IF SET  
    MOV #PUPA,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,ASFR ;CLEAR THE DEVICE  
    GETPAS,REGIN,REGPA ;GET PHYSICAL ADDRESS FROM 16-BIT PUPA  
    MOV #16,VECTRD ;LOAD INT. VECTRD  
    MOVR #RUPA,RO ;LOAD BR LEVEL  
    MOVR #RUEPA,RO ;SET BUFFER POINTER  
    BTC #3,RO ;CLEAR EXT. MEMORY BITS  
    MOVR #RUEPA,R1 ;GET EXTENDED MEMORY BITS  
    ASR R1  
    ASR R1  
    ASR R1  
    ASR R1  
    ASR R1  
    ASR #177774,R1 ;CLEAR OFF EXCESS  
    ADD R1,RO  
    MOV R1,ROFF ;LOAD COMBINED ADDRESS  
    MOV #10, #WCR ;LOAD HIGH 16 BITS OF Z  
    MOVR #Q,QBAR ;LOAD LOW 16 BITS OF Z  
    MOV #RT15122,ACSR ;SET RES. AND MODE AND CELL INT. ENABLE  
    BTIS #TEST7,ASFR ;SET 7 PULSES  
    BTIS #RT10,ACSR ;ENABLE THE DEVICE  
    EXIT, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.  
  
    RIS #ENDDMA,8SF ;STOP TRANSERS  
    RIS #CLRALL,ASFR ;CLEAR NCV11  
  
    PIROS,REGIN,11S ;QUEUE UP TO CONTINUE AT 11S AND RTI  
  
    MOV VECTBL,VECTRA ;RESET THE VECTOR  
    CLR VECTRA ;RESET THE VECTOR  
    NOP  
    NOP  
    NOP
```

NCRA DEC/X11 SYSTEM EXERCISER MODULE
XNCRA0.P11 15-NOV-78 11:40

MACY11 30A(1052) 22-NOV-78 11:22 PAGE 17

SEQ 0016

```

713
719
720
721
722 003400*
723 003401 104407 000000*
724 003401 104407 000000*
725 003410 104413 000000*
726 003414* 000167 174670
727
728
729
730 003420* 000J24
731
732
733
734
735 003470* 000240
736 003472* 000240
737 003474* 000240
738 003476* 000240
739 003500* 000400
740 004500* 000400
741 005500* 000000
742
743 000001
*****END OF PASS*****
*****TST21: *****  

    BREAKS, BEGIN ;TEMPORARY RETURN TO MONITOR.  

    ENDITS, BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  

    SIGNAL END OF ITERATION.  

    MONITOR SHALL TEST END OF PASS  

    ;RUN AGAIN
1$: JMP LOGIC
;MODULE PATCH SPACE
    .RLKW 20.
;MODULE READ BUFFER SPACE
RUF1: NOP ;NOP'S MUST REMAIN <NOT A PATCH AREA>
RUF2: NOP
RUF3: .RLKW 256.
RUF4: 0 .RLKW 256.
RUF5: .END

```

NCRA DEC/X11 SYSTEM EXERCISER MODULE
XNCRA0.P11 15-NOV-78 11:40

MACY11 30A(1052) 22-NOV-78 11:22 PAGE 19

SEQ 0017

| | 000000 | 210# | 218# | 219# | 220# | 221# | 222# | 223# | 224# | 225# | 226# | 227# | 337# | 355* |
|--------|----------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| ACSR | 00012P | 195# | 274* | 275* | 276 | 295* | 292 | 309* | 321 | 315* | 321 | 336* | 432 | 434 |
| | | 353 | 372* | 345* | 388 | 390 | 399* | 407* | 411 | 412 | 421* | 429* | 501* | 521* |
| | | 443* | 453* | 441 | 466* | 468 | 473* | 475 | 480* | 482 | 484* | 489 | 501* | 521* |
| | | 657* | 648 | 654* | 644 | 544 | 585* | 594* | 595* | 596 | 602* | 603 | 648* | 656* |
| ADDR | 000006P | 151# | 237 | 242 | 244 | 251 | | | | | | | | |
| ADDR22 | 001000 | 209# | | | | | | | | | | | | |
| ADM | 0001264P | 223# | 263 | | | | | | | | | | | |
| ASB | 000106P | 149# | | | | | | | | | | | | |
| ASTAT | 000104P | 187# | 273* | 276 | 296* | 297 | 311* | 312 | 320* | 321 | 335* | 337 | 357* | 358 |
| | | 373* | 349* | 390 | 411* | 412 | 433* | 434 | 466* | 467* | 467* | 469 | 474* | 475 |
| | | 441* | 442* | 448* | 486 | 505* | 523* | 524 | 533* | 534 | 543* | 544 | 586* | 587* |
| | | 551* | 552 | 621* | 603 | 640* | 641* | 649 | 655* | 658 | 663* | 665 | | |
| AVECT1 | 000000 | 211# | 228 | 229 | 230 | 231 | | | | | | | | |
| AVMS | 000112P | 205# | | | | | | | | | | | | |
| RAD1 | 0005609P | 221# | 261 | 410* | 411 | 455* | 474 | 573* | 593 | 635* | 655 | 702* | | |
| RAD2 | 000270P | 225# | 264 | | | | | | | | | | | |
| BEGIN | 000005P | 148# | 256 | 257 | 269 | 270 | 279 | 292 | 283 | 291 | 292 | 300 | 306 | 307 |
| | | 315 | 124 | 330 | 331 | 340 | 343 | 344 | 352 | 353 | 361 | 368 | 369 | 376 |
| | | 383 | 394 | 393 | 397 | 398 | 405 | 406 | 415 | 419 | 420 | 427 | 428 | 437 |
| | | 441 | 442 | 449 | 450 | 464 | 471 | 478 | 485 | 492 | 499 | 500 | 508 | 515 |
| | | 516 | 528 | 537 | 547 | 553 | 554 | 560 | 561 | 564 | 580 | 581 | 590 | 599 |
| | | 606 | 613 | 614 | 620 | 621 | 524 | 524 | 545 | 552 | 661 | 668 | 676 | 677 |
| RIT0 | 00000C1 | 822# | 528 | 638 | 428 | 429 | 421 | 541 | 714 | 722 | 637 | 678 | 705 | |
| RIT1 | 0000062 | 520# | 497 | 520 | 429 | 422 | 421 | 541 | 714 | 722 | | | | |
| RIT10 | 0020005 | 209# | | | | | | | | | | | | |
| RIT11 | 0024000 | 229# | 217 | | | | | | | | | | | |
| RIT12 | 0010000 | 220# | 570 | 271 | | | | | | | | | | |
| RIT13 | 0020000 | 209# | | | | | | | | | | | | |
| RIT14 | 0040000 | 209# | | | | | | | | | | | | |
| RIT15 | 1000002 | 229# | | | | | | | | | | | | |
| RIT16 | 0000004 | 209# | | | | | | | | | | | | |
| RIT17 | 0000010 | 209# | | | | | | | | | | | | |
| RIT18 | 0000040 | 209# | | | | | | | | | | | | |
| RIT19 | 0000000 | 209# | | | | | | | | | | | | |
| RIT20 | 0000000 | 209# | | | | | | | | | | | | |
| RIT21 | 0000000 | 209# | | | | | | | | | | | | |
| RIT22 | 0000000 | 209# | | | | | | | | | | | | |
| RIT23 | 0000000 | 209# | | | | | | | | | | | | |
| RIT24 | 0000000 | 209# | | | | | | | | | | | | |
| RIT25 | 0000000 | 209# | | | | | | | | | | | | |
| RIT26 | 0000000 | 209# | | | | | | | | | | | | |
| RIT27 | 0000000 | 209# | | | | | | | | | | | | |
| RIT28 | 0000000 | 209# | | | | | | | | | | | | |
| RIT29 | 0000000 | 209# | | | | | | | | | | | | |
| RIT30 | 0000000 | 209# | | | | | | | | | | | | |
| RIT31 | 0000000 | 209# | | | | | | | | | | | | |
| RIT32 | 0000000 | 209# | | | | | | | | | | | | |
| RIT33 | 0000000 | 209# | | | | | | | | | | | | |
| RIT34 | 0000000 | 209# | | | | | | | | | | | | |
| RIT35 | 0000000 | 209# | | | | | | | | | | | | |
| RIT36 | 0000000 | 209# | | | | | | | | | | | | |
| RIT37 | 0000000 | 209# | | | | | | | | | | | | |
| RIT38 | 0000000 | 209# | | | | | | | | | | | | |
| RIT39 | 0000000 | 209# | | | | | | | | | | | | |
| RIT40 | 0000000 | 209# | | | | | | | | | | | | |
| RIT41 | 0000000 | 209# | | | | | | | | | | | | |
| RIT42 | 0000000 | 209# | | | | | | | | | | | | |
| RIT43 | 0000000 | 209# | | | | | | | | | | | | |
| RIT44 | 0000000 | 209# | | | | | | | | | | | | |
| RIT45 | 0000000 | 209# | | | | | | | | | | | | |
| RIT46 | 0000000 | 209# | | | | | | | | | | | | |
| RIT47 | 0000000 | 209# | | | | | | | | | | | | |
| RIT48 | 0000000 | 209# | | | | | | | | | | | | |
| RIT49 | 0000000 | 209# | | | | | | | | | | | | |
| RIT50 | 0000000 | 209# | | | | | | | | | | | | |
| RIT51 | 0000000 | 209# | | | | | | | | | | | | |
| RIT52 | 0000000 | 209# | | | | | | | | | | | | |
| RIT53 | 0000000 | 209# | | | | | | | | | | | | |
| RIT54 | 0000000 | 209# | | | | | | | | | | | | |
| RIT55 | 0000000 | 209# | | | | | | | | | | | | |
| RIT56 | 0000000 | 209# | | | | | | | | | | | | |
| RIT57 | 0000000 | 209# | | | | | | | | | | | | |
| RIT58 | 0000000 | 209# | | | | | | | | | | | | |
| RIT59 | 0000000 | 209# | | | | | | | | | | | | |
| RIT60 | 0000000 | 209# | | | | | | | | | | | | |
| RIT61 | 0000000 | 209# | | | | | | | | | | | | |
| RIT62 | 0000000 | 209# | | | | | | | | | | | | |
| RIT63 | 0000000 | 209# | | | | | | | | | | | | |
| RIT64 | 0000000 | 209# | | | | | | | | | | | | |
| RIT65 | 0000000 | 209# | | | | | | | | | | | | |
| RIT66 | 0000000 | 209# | | | | | | | | | | | | |
| RIT67 | 0000000 | 209# | | | | | | | | | | | | |
| RIT68 | 0000000 | 209# | | | | | | | | | | | | |
| RIT69 | 0000000 | 209# | | | | | | | | | | | | |
| RIT70 | 0000000 | 209# | | | | | | | | | | | | |
| RIT71 | 0000000 | 209# | | | | | | | | | | | | |
| RIT72 | 0000000 | 209# | | | | | | | | | | | | |
| RIT73 | 0000000 | 209# | | | | | | | | | | | | |
| RIT74 | 0000000 | 209# | | | | | | | | | | | | |
| RIT75 | 0000000 | 209# | | | | | | | | | | | | |
| RIT76 | 0000000 | 209# | | | | | | | | | | | | |
| RIT77 | 0000000 | 209# | | | | | | | | | | | | |
| RIT78 | 0000000 | 209# | | | | | | | | | | | | |
| RIT79 | 0000000 | 209# | | | | | | | | | | | | |
| RIT80 | 0000000 | 209# | | | | | | | | | | | | |
| RIT81 | 0000000 | 209# | | | | | | | | | | | | |
| RIT82 | 0000000 | 209# | | | | | | | | | | | | |
| RIT83 | 0000000 | 209# | | | | | | | | | | | | |
| RIT84 | 0000000 | 209# | | | | | | | | | | | | |
| RIT85 | 0000000 | 209# | | | | | | | | | | | | |
| RIT86 | 0000000 | 209# | | | | | | | | | | | | |
| RIT87 | 0000000 | 209# | | | | | | | | | | | | |
| RIT88 | 0000000 | 209# | | | | | | | | | | | | |
| RIT89 | 0000000 | 209# | | | | | | | | | | | | |
| RIT90 | 0000000 | 209# | | | | | | | | | | | | |
| RIT91 | 0000000 | 209# | | | | | | | | | | | | |
| RIT92 | 0000000 | 209# | | | | | | | | | | | | |
| RIT93 | 0000000 | 209# | | | | | | | | | | | | |
| RIT94 | 0000000 | 209# | | | | | | | | | | | | |
| RIT95 | 0000000 | 209# | | | | | | | | | | | | |
| RIT96 | 0000000 | 209# | | | | | | | | | | | | |
| RIT97 | 0000000 | 209# | | | | | | | | | | | | |
| RIT98 | 0000000 | 209# | | | | | | | | | | | | |
| RIT99 | 0000000 | 209# | | | | | | | | | | | | |
| RIT100 | 0000000 | 209# | | | | | | | | | | | | |
| RIT101 | 0000000 | 209# | | | | | | | | | | | | |

NCBA DEC/X11 SYSTEM EXEC SER MODULE MACV11 30A(1052) 22-NOV-78 11:22 PAGE 20
 XNCBAU.P11 15-NOV-78 11:40 CROSS REFERENCE TABLE -- USER SYMBOLS SEQ 0018
 CLRALL = 004000 217# 235 294 371 387 409 431 452 494 502 534 517 542
 CONFIG 000056R 557 582 617 643 654 687 705 721 744 762 784 803 821
 CSR 002522R 173# 218# 236 258 272* 273 293* 296 303* 312* 317* 321 337*
 CSRA 000100R 488 519# 522* 523 533 541* 543 555* 567* 577* 584 595*
 CSRB 000227R 183# 226# 240 251* 269# 279# 299# 319*
 DATAERS = 1044004 209#
 DATESRS = 000014R 155#
 DNDMA = 00400 213# 226 251 273 293# 306 321# 337# 353# 371# 388# 395#
 ENDITS = 104413 209# 226 251 273 293# 306 321# 337# 353# 371# 388# 395#
 ENDS = 104410 209# 226 251 273 293# 306 321# 337# 353# 371# 388# 395#
 ERRTYPE = 000106R 188#
 EXITS = 104400 209# 226 251 273 293# 306 321# 337# 353# 371# 388# 395#
 FREE = 000150R 206#
 GETFAS = 104415 209# 226 251 273 293# 306 321# 337# 353# 371# 388# 395#
 GWRDPS = 000044R 103# 168#
 HNRDERS = 104405 209# 226 251 273 293# 306 321# 337# 353# 371# 388# 395#
 HRDPAS = 000050R 478 485 492 508 528 537 547 557 567 577 584 595 604
 ICOUNT = 000036R 170# 165#
 ICOUNT = 000040R 166#
 IDNOM = 000012R 195#
 IDMDX = 000000 207#
 INTR = 000030R 164#
 INTR = 000030R 164#
 JOY = 000366R 224#
 LOGIC = 000310R 225#
 MAP22S = 104416 209# 727
 MODNAM = 000000R 149#
 MODSP = 000252R 163 207#
 MSGNS = 104403 209#
 MSGSS = 104402 209#
 MSGS = 104401 209#
 NULL = 000000 209#
 OFF = 000254R 479 486 492 508 528 537 547 557 567 577 584 595 604
 OPEN = 000000 219# 259 422* 433 453* 460 503* 505 571* 586 631* 640 700*
 PTOAS = 104420 209#
 PASCNT = 000034R 164#
 POPQS = 000004 209#
 POPSP = 000326 209#
 POPSP2 = 000266 209#
 PRTF = 000000 209#
 PRTVO = 000000 154 209#
 PRTV1 = 000040 209#
 PRTV2 = 000100 209#
 PRTV3 = 000140 209#
 PRTV4 = 000200 209#
 PRTV5 = 000240 209#
 PRTV6 = 000300 153 209#
 PRTV7 = 000340 209#

NCBA DEC(X11 SYSTEM EXERCISER MODULE MACY11 30A(1052) 22-NOV-78 11:27 PAGE 21
 XNCBAO.P11 15-NOV-78 11:40 CROSS REFERENCE TABLE -- USER SV4BOLS SEQ 0019

| | | |
|---------|-------------|------|
| PS | = 177776 | 209# |
| PSW | = 177776 | 209# |
| PUSH | = 005746 | 209# |
| PUSH2 | = 024646 | 209# |
| RANDS | = 104417 | 209# |
| RANUPM | = 00000540R | 109# |
| RBUPPA | = 00012260R | 109# |
| RBUPPA | = 00012260R | 109# |
| RBUPS2 | = 00012220R | 109# |
| RBUPVA | = 00012240R | 109# |
| RESTR1 | = 000310R | 191 |
| RES1 | = 000056R | 174# |
| RES2 | = 000060R | 175# |
| RSTRT | = 000112R | 191# |
| SBADR | = 000102R | 184# |
| SFR | = 000262R | 227# |
| | | 235* |
| | | 457* |
| | | 577* |
| | | 578* |
| | | 579* |
| | | 592* |
| | | 617* |
| | | 624 |
| | | 635 |
| | | 656 |
| | | 691 |
| SFRHB | = 000274R | 227# |
| SOFCNT | = 000042R | 157# |
| SOFERS= | 104406 | 209# |
| SOPFAS | = 000046R | 169# |
| SPPOINT | = 000032R | 163# |
| SPSIZEZ | = 000040 | 1# |
| SR1 | = 000016R | 150# |
| SR2 | = 000020R | 157# |
| SR3 | = 000024R | 159# |
| SR4 | = 000024R | 159# |
| START | = 000310R | 167# |
| STAT | = 000026R | 161# |
| SVR0 | = 000062R | 176# |
| SVR1 | = 000064R | 177# |
| SVR2 | = 000066R | 178# |
| SVR3 | = 000070R | 179# |
| SVR4 | = 000072R | 180# |
| SVR5 | = 000074R | 181# |
| SVR6 | = 000076R | 182# |
| SVSCNT | = 000052R | 127# |
| TEST2 | = 000002 | 214# |
| TRPFD= | = 000022 | 200# |
| TSTCON= | = 000004 | 219# |
| TSTDMA= | = 000010 | 219# |
| TSTI1 | = 00406R | 255# |
| TSTI10 | = 001204R | 374 |
| TSTI11 | = 001300R | 392# |
| TSTI12 | = 001300R | 404# |
| TSTI13 | = 001300R | 426# |
| TSTI14 | = 001470R | 440# |
| TSTI15 | = 002554R | 429# |
| TSTI16 | = 002536R | 514# |
| TSTI17 | = 002572R | 545 |
| TST2 | = 00452R | 557# |
| TST20 | = 003140R | 556# |
| TST21 | = 003400R | 616 |
| TST3 | = 000562R | 679 |
| | | 675# |
| | | 722# |
| | | 790# |
| | | 812# |
| | | 818# |
| | | 843* |
| | | 854* |
| | | 867* |
| | | 873 |
| | | 897* |
| | | 914* |
| | | 957* |
| | | 979* |

NCBA DEC/X11 SYSTEM EXERCISE MODULE MACV11 30A(1052) 22-NOV-78 11:22 PAGE 22
XNCBAD.P11 15-NOV-78 11:40 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0020

```
ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0
XNCBAO,XNCRAO/SOL/CRF:SYM=DDXCOM,XNCRAO
RUN-TIME: 2 3 .4 SECONDS
RUN-TIME RATIO: 33/55=.7
CORE USED: 7K (13 PAGES)
```