

PAAF DEC/X11 SYSTEM EXERCISER MODULE
XPAAAF0.P11 12-OCT-78 12:03

MACY11 30A(1052) 12-OCT-78 16:55 PAGE 2

SEQ 0001

.REM -

IDENTIFICATION

PRODUCT CODE: AC-E86CF-MC
PRODUCT NAME: CXPAAF0 PA611 READER 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 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) 1973, 1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT:

PAA IS AN IOMOD THAT EXERCISES UP TO 16 HIGH SPEED PAPER TAPE READERS BY READING A STANDARD BINARY COUNT PATTERN. IT REPORTS ALL ERRORS ON THE CONSOLE ITTY. THE MODULE IS DESIGNED TO ACTIVATE ALL DEVICES AND READ WITH ALL ACTIVE DEVICES RUNNING CONCURRENTLY.

2. REQUIREMENTS:

HARDWARE: AT LEAST ONE PA611-R CONTROL UNIT AND ONE PR69E HIGH SPEED READER

STORAGE:: PAA REQUIRES:

1. DECIMAL WORDS: 404
2. OCTAL WORDS: 0624
3. OCTAL BYTES: 1450

3. PASS DEFINITION:

ONE PASS OF THE PAA MODULE CONSISTS OF READING 2048 CHARACTERS

4. EXECUTION TIME:

PAA RUNNING ALONE ON A PDP11/05 PROCESSOR TAKES APPROXIMATELY---MINUTES TO COMPLETE ONE PASS.

5. CONFIGURATION REQUIREMENTS:

DEFAULT PARAMETERS:

DEVADR: 172600, VECTOR: 300, RPI: 4, DEVCNT: 1

REQUIRED PAPANETERS:

NONE

6. DEVICE/OPTION SET-UP:

- A: TURN ON ALL PR68E READERS TO BE TESTED
B: LOAD EACH READER WITH A BINARY COUNT TAPE AND POSITION THE TAPE SUCH THAT A NON-ZERO PUNCH IS OVER THE READER PHOTO CELLS (A FABRICATED TEST LOOP TAPE IS RECOMMENDED TO ELIMINATE THE NEED TO RELOAD)

7. MODULE OPERATION:

TEST SEQUENCE:

- A. TEST UP TO 16 POSSIBLE DEVICES FOR SELECTION
- B. STORE THE DEVICE NO. OF THOSE DEVICES SELECTED
TO BE TESTED AND SET UP THEIR APPROPRIATE VECTOR
- C. TURN ON THE INTERRUPT ENABLE AND READER ENABLE FOR
ALL ACTIVE DEVICES.
- D. INTERRUPT SERVICE
 - 1. READ 64 CHARACTERS TO SYNCHRONIZE THE DATA PATTERN
 - 2. REORDER TESTING FOR ERRORS
 - 3. TEST FOR AND REPORT ANY ERROR BITS IN THE CSR
 - 4. COMPARE THE DATA READ WITH THE EXPECTED DATA
 - 5. IF NO DATA ERRORS GO TO STEP 7
 - 6. IF DATA ERROR RESYNC THE DATA
 - 7. ENABLE THE READER AND INTERRUPT TO READ THE NEXT
FRAME
 - 8. REPEAT STEPS 2-7 UNTIL 2096. FRAMES HAVE BEEN
PROCESSED
- E. TURN OFF ALL ACTIVE DEVICES AND REPORT END OF PASS
- F. RESTART AT STEP C

JSR TABLE:

TO LINK THE INDIVIDUAL INTERRUPTS WITH THE SERVICE ROUTINES
THESE JSR TABLE CONTAINING IC ENTRIES, EACH DEVICE
VECTOR IS SET UP TO POINT TO A SERVICE ROUTINE WITHIN THE TABLE
WHICH TRANSFERS CONTROL TO THE SERVICE ROUTINE AND POINTS
RS TO AN OFFSET THAT THE SERVICE ROUTINE WILL USE TO GEN-
ERATE THE CORRESPONDING REGISTER ADDRESS AND DATA TABLE
ENTRY FOR THAT LINE.

FIFO QUEUE:

TO ALLOW THE SERVICE ROUTINES TO USE THE SAME GPRS FOR
SERVICING UP TO 16 CONCURRENT INTERRUPTS FROM DIFFERENT
LINES THE INTERRUPT REQUEST IS STORED IN A FIFO QUEUE TO
PREVENT SERVICE RUTINE STORES HAS 16 ENTRIES AND THE INT-
ERRUPT QUEUE AND CALL IS SERVICE THE OFFSET AND THE LINE
WHEN THE "PIRQ" CALL IS EXECUTED A "PIRQ" CALL.
FROM THE QUEUE AND THE SERVICE. TWO POINTERS ARE
REQUIRED TO SERVICE THE QUEUE:

QPTR1 USED TO STORE THE OFFSET
QPTR2 USED TO RETRIEVE THE OFFSET

ERROR RETURN QUEUE:

WHEN AN "ERROR" OR A "DATER" CALL IS EXECUTED THE MONITOR
QUEUES THE RETURN ADDRESS IN ITS "IDQUE". SINCE IT IS
POSSIBLE THAT ANOTHER LINE MAY ENJOY A HIGHER PRIORITY IN
THE MONITOR'S QUEUE, IT COULD GET CONTROL OF THE MODULE'S
PIRQ SERVICE ROUTING AND CLOBBER THE REGISTERS WHEN THE
ERROR LINE RETURNS. IT ENDS UP WITH THE WRONG REGISTER
ADDRESS. TO PREVENT THIS FROM HAPPENING, A SECOND FIFO
QUEUE IS USED TO QUEUE RETURNS FROM ERRORS. PRIOR TO THE
ANY ERROR CALL THE INFORMATION NECESSARY TO RESTORE THE
CORRECT ADDRESS IS STORED IN THIS QUEUE AND WHEN CONTROL
IS RETURNED AFTER THE ERROR THIS INFORMATION IS RETRIEVED.
THE QUEUE CONSISTS OF 16 BYTE ENTRIES AND REQUIRES THE
FOLLOWING POINTERS FOR ITS OPERATION:

REQP1 SAVES R1 IN THE QUEUE
REQP2 RETRIEVES R1 FROM THE QUEUE

OPERATION OPTIONS:

8.

A. LOCATION DVID1 MAY BE ALTERED TO CHANGE THE NUMBER
AND COMBINATION OF DEVICES TO BE TESTED.
R10=DEV0, B111=DEV1.....B115=DEV15.

9.

NON-STANDARD PRINTOUTS:

NONE: ALL PRINTOUTS HAVE THE STANDARD MEANING AND FORMAT
DESCRIBED IN THE DEC/X11 DOCUMENT

3PA611R DEC/X11 EXERCISER MODULE

```

000000- IOMOD <PAAF> 172600 300 4 2048 53
000000- MODULE 140000 <PAAF> 172600 300 4 2048 53
          TITLE PAAF DEC/X11 SYSTEM EXERCISER MODULE
          DDXCOM VERSION 6 23-NOV-78
          LIST BIN
*****
000000- REGTN: *****
000005- 040520 043101 040 MODNAM: /ASCII /DAAF /MODULE NAME
000006- 000005- 000005- 000005- 000005- 000005- 000005- 000005- 000005- 000005-
          USED TO KEEP TRACK OF WBUF USAGE
000006- 172600 ADDR: 172600+0 ;1ST DEVICE ADDR
000010- 000300 VECTCR: 300+C ;1ST DEVICE VFCTOR
000013- 000000 R1: RYTE PDTV4+0 ;1ST RR LEVEL
000014- 000000 R2: RYTE PDTV+0 ;2ND RR LEVEL
000016- 000000 DVID1: +1 ;DEVICE INDICATOR 1
000022- 000000 SR1: OPEN ;SWITCH REGISTER 1
000022- 000000 SR2: OPEN ;SWITCH REGISTER 2
000022- 000000 SR3: OPEN ;SWITCH REGISTER 3
000024- 000000 SR4: OPEN ;SWITCH REGISTER 4
*****
000026- 140300 STAT: 140UCC ;STATUS WORD
000030- 000324 SPOINT: START ;MODULE START ADDR
000032- 000324 MODSP: MODSP ;MODULE STACK POINTER
000034- 000000 PASCNT: C ;PASS COUNT
000040- 000000 ICONF: 2C49 ;% OF ITERATIONS PER PASS=2048
000042- 000000 SOFCNT: C ;LOC TO COUNT ITERATIONS
000044- 000000 HRDCNT: C ;LOC TO SAVE TOTAL SOFT ERRORS
000048- 000000 SOFPAC: C ;LOC TO SAVE TOTAL HARD ERRORS
000052- 000000 HRDPAS: C ;LOC TO SAVE SOFT ERRORS PER PASS
000054- 000000 RANUM: C ;LOC TO SAVE HARD ERRORS PER PASS
000056- 000000 CONFIG: C ;% OF SYS ERRORS ACCUMULATED
000060- 000000 RES1: C ;WORDS RANDOM WHEN RAND MACRO IS CALLED
000062- 000000 SVRA: C ;RESERVED FOR MONITOR USE
000064- 000000 SVR1: OPEN ;RESERVED FOR MONITOR USE
000066- 000000 SVR2: OPEN ;LOC TO SAVE R0
000068- 000000 SVR3: OPEN ;LOC TO SAVE R1
000070- 000000 SVR4: OPEN ;LOC TO SAVE R2
000072- 000300 SVR5: OPEN ;LOC TO SAVE R3
000074- 000300 SVR6: OPEN ;LOC TO SAVE R4
000076- 000000 SVR7: OPEN ;LOC TO SAVE R5
000078- 000000 SVR8: OPEN ;LOC TO SAVE R6
000102- 000000 CSRA: OPEN ;ADDR OF CURRENT CSR
000104- 000000 ACSR: OPEN ;ADDR OF GOOD DATA, OR
000106- 000000 WASADR: OPEN ;CONTENTS OF CSR
000108- 000000 ASTAT: OPEN ;ADDR OF BAD DATA OR
000110- 000000 ERPTVP: OPEN ;STATUS REG CONTENTS
000112- 000000 ASB: OPEN ;TYPE OF ERROR
000114- 000000 AWAS: OPEN ;ACCEPTED DATA
          RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
          WDTO: OPEN ;WORDS TO MEMORY PER ITERATION

```

```

000116- 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000120- 000000 INTR: OPEN ;% OF INTERRUPTS PER ITERATION
000122- 000353 IDNUM: 53 ;MODULE IDENTIFICATION NUMBER=53
          .REFT SPSIZ ;MODULE STACK STARTS HERE.
          .MLIST
          .WORD C
          .LIST
          .ENDP
MODSP: *****
*****
;MODULE INITIALIZATION
START: MOV #2048,INTR ;2048 INTPS
      MOV #1024,WDT0 ;1024 WORDS TO
RFSTR: MOV ICONF,COUNT ;DO APPROX. 2048 INTPS/PASS
      MOV #2048,COUNTC ;KEEP TRACK OF READS ISSUED
      CLP CFTOUT ;RESET END OF PASS FLAG
      MOV #0,OPTR1 ;SET UP SERVICE QUEUE POINTERS
      MOV #0,OPTR2
      MOV #0,PEOP1 ;SET UP ERROR RETURN QUEUE POINTERS
      MOV #0,PEOP2
      MOV DVID1,R0 ;GET DEVICE SELECTION PARAMETER
      MOV ENFS,REGTN ;RR IF ANY DEVICES SELECTED FOR TEST
      CLR P1 ;R1 AND R2 TO START AT ZERO
      CLR P2
      CLR P3
      CLR P4
      MOV VECTOR,R3 ;START AT BEGINNING OF VECTOR AREA
      MOV #JSRTAR,R4 ;START AT BEGINNING OF JSR TABLE
      INC R1 ;COUNT A DEVICE
      ASR PC ;SHIFT SELECT BIT INTO "C"
      ADD #4,R3 ;RR IF SELECT BIT IS ON A "1"
      AND #4,R4 ;UPDATE VECTOR POINTER
      CMP #17,R1 ;UPDATE JSR TABLE POINTER
      STMP R0 ;LAST POSSIBLE DEVICE
      RFO R2 ;GO START UP ACTIVE DEVICES IF YES
      RFO R3 ;GO TEST FOR NEXT ONE
      RFO R4 ;SET UP PWR PTR
      RFO R5 ;SET UP PRIORITY LEVEL
      RFO R6 ;UPDATE JSR TABLE POINTER
      RFO R7 ;COUNT ACTIVE DEVICE
      RFO R8 ;STORE ACTIVE DEV. NO. IN TABLE
      RFO R9 ;GO TEST FOR LAST DEVICE
;DEVICE AND DATA TABLES
DEVTAR: .PLKW R. ;16 RYTE ACTIVE DEVICE TABLE
DATPAR: .PLKW B. ;16 RYTE DATA TABLE
JSRTAR: JSR P5,PINTR ;LINK TO READER 0

```

```

298 000456 000000 C
299 000460 004567 000176 JSP R5,RINTR ;LINK TO READER 1
300 000464 000000 000170 JSR R5,RINTR ;LINK TO READER 2
301 000468 004567 000162 JSP R5,RINTR ;LINK TO READER 3
302 000472 000010 000154 JSR R5,RINTR ;LINK TO READER 4
303 000476 004567 000146 JSP R5,RINTR ;LINK TO READER 5
304 000480 000020 000140 JSR R5,RINTR ;LINK TO READER 6
305 000484 004567 000132 JSP R5,RINTR ;LINK TO READER 7
306 000488 000030 000124 JSR R5,RINTR ;LINK TO READER 10
307 000492 004567 000116 JSP R5,RINTR ;LINK TO READER 11
308 000496 000040 000110 JSR R5,RINTR ;LINK TO READER 12
309 000500 004567 000102 JSP R5,RINTR ;LINK TO READER 13
310 000504 000050 000074 JSR R5,RINTR ;LINK TO READER 14
311 000508 004567 000066 JSP R5,RINTR ;LINK TO READER 15
312 000512 000060 000060 JSR R5,RINTR ;LINK TO READER 16
313 000516 004567 000052 JSR R5,RINTR ;LINK TO READER 17
314 000520 000070 000074 JSP R5,RINTR ;LINK TO READER 17

```

INITIAL START UP ROUTINES

```

000612 010267 000630 STUP: MOV R2,ACTDEV ;SAVE ACTIVE DEVICE COUNT
000616 012767 000100 ;R2 TO INDICATE 64 CHARS. READ
000620 116200 000412 000606 MOVR R2,SYNFLG ;GET AN ACTIVE DEVICE NO.
000624 110001 MOVR R0,R1 ;DUPLICATE DEVICE NO. IN R1
000628 000000 ASL R0 ;GENERATE CSR ADDRESS IN R0
000632 000000 ADD ADDR,R0
000636 066700 PIS #101,(R0) ;GO READ A CHAR. AND INTERRUPT
000640 052710 DEC COUNTC ;COUNT A READ ISSUED
000644 005367 DEC BPL 15 ;COUNT 15
000648 100363 EXITS,REGIN ;ARE KICKED OFF
000652 104400 000000 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

;READER INTERRUPT SERVICE ROUTINES
;ALL INTERRUPTS ENTER HERE VIA JSR TABLE LINKAGE
RINTR: MOVR (R5),QPTR1 ;STORE REQUEST IN FIFO QUEUE
INC QPTR1 ;UPDATE Q POINTER

```

```

354 000672 022767 001312 000534 CMP #RQ+20,QPTR1 ;QUEUE POINTER BEYOND LIMIT
355 000700 001023 RNE 15 ;RR IF NOT
356 000702 012767 001272 000524 MOVR #RQ,QPTR1 ;RESET QUEUE POINTER
357 000710 012605 MOVR (R5),R5 ;CLEAN UP STACK
358 000712 00C004 000000 000720 ;-----
;PIQS,REGIN,RDSERV ; QUEUE UP TO CONTINUE AT RDSERV AND RTI
;-----

;ROUTINE TO SERVICE FIFO QUEUE
RDSERV: TST GFTOUT ;END OF PASS?
RMI FNPS ;RR IF YES
MOVR QPTR2,R0 ;GET OFFSET FROM Q
MOVR QPTR1,R1 ;DUPLICATE OFFSET IN R1
INC QPTR1 ;UPDATE Q POINTER
CMP #RQ+20,QPTR2 ;QUEUE POINTER BEYOND LIMIT
RNE 15 ;RR IF NOT
MOVR RQ,QPTR2 ;RESET POINTER
ASR R1 ;GENERATE DEVICE NO. IN R1
ADD ADDR,R0 ;GEN. CSR ADDRESS IN R0
TST SYNFLG ;DATA TABLE SYNCHRONIZED?
RNE 15 ;RR IF YES (64 INTERRUPTS)
MOVR 2(R0),DATTAR(R1) ;LOAD CVR,DRR INTO DATA TABLE
DEC SYNFLG ;COUNT IT
INC (R0) ;READ ANOTHER FRAME
EXITS,REGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

2S: TST (R0) ;ERROR BIT IN CSR?
RNE 15 ;RR IF NONE
MOVR #RSER0,FORK ;SET UP RETURN TO RSER0
RFRD (R0) ;GO REPORT THE ERROR
TSTR (R0) ;NONE SET?
RNE 15 ;RR IF SET
MOVR #RSER1,FORK ;SET UP RETURN TO RSER1
RFRD (R0) ;GO REPORT THE ERROR
TSTR (R0) ;DID DATA IN DRR=DATA IN TABLE?
RNE 15 ;RR IF YES
MOVR DATTAR(R1),DATTAR(R1) ;RUMP THE DATA CHAR. +1
RFRD (R0),DATTAR(R1) ;DID DATA IN DRR=DATA IN TABLE?
RNE 15 ;RR IF YES
RFRD DATTAR ;GO REPORT THE ERROR
COUNT ;COUNT THE INTERRUPT
RNE 15 ;RR IF NOT 2048 YET
CMV GFTOUT ;SET END OF PASS FLAG
RFRD FNPS ;GO TURN OFF ALL ACTIVE READERS

1S: ENDDITS,REGIN ;SIGNAL END OF ITERATION
TST COUNTC ;MONITOR SHOULD TEST END OF PASS
RNE 15 ;HAVE WE ISSUED ENOUGH READS?
RNE 15 ;RR IF YES

INC (R0) ;READ ANOTHER FRAME
DEC COUNTC ;COUNT ANOTHER READ ISSUED

EX: EXITS,REGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

;ROUTINE TO TURN OFF ALL ACTIVE READERS AFTER 2048. TRANSFERS

```

410 001122 016701 000320
 411 001126 116100 000412
 412 001134 006300
 413 001134 006300
 414 001136 066700 176644
 415 001142 005310
 416 001148 005300
 417 001148 100361
 418 001150 104413 000000
 419
 420
 421
 422
 423 001154 010067 176720
 424 001160 011067 176716
 425 001164 004767 000142
 426 001170 012767 000011 176710
 427
 428 001176 104405 000000 000000
 429
 430 001204 004767 000152
 431 001210 000194 000230
 432
 433 001214 010067 176660
 434 001220 005720 176662
 435 001225 111024 176662
 436 001226 116167 000432 176652
 437 001234 010067 176644
 438 001240 025767 000432
 439 001250 010167 176626
 440
 441 001254 111011
 442 001256 005740
 443
 444 001260 104404 000000
 445
 446 001264 004767 000072
 447 001270 000634
 448
 449 001272 000010
 450 001312 000010
 451
 452
 453
 454 001332 110177 000132
 455 001336 005267 000076
 456 001342 02767 001332 000070
 457 001356 04767
 458 001352 012767 001312 000060
 459 001360 000207
 460
 461 001362 117701 000054
 462 001366 005267 000050
 463 001372 02767 001332 000042
 464 001400 001003
 465 001402 012767 001312 000032

ENPS: MOV ACTDEV,R1 ;GET NO OF ACTIVE DEVICES
 1S: MOV DVTAR(R1),R0 ;GET ACTIVE DEVICE NO. FM TABLE
 ASL R0 ;GEN. CSR ADDRESS
 ASL R0
 ADD ADDR,R0
 CLP (R0) ;TURN OFF READER
 DEC R1 ;COUNT IT
 RPL 1S ;RR UNTIL ALL OFF
 ENDTIS,REGIN ;SIGNAL END OF ITERATION
 ;MONITOR SHALL TEST END OF PASS
 ;TO SHUT OFF INTERRUPTS
 ;ERRPR REPORTING ROUTINES
 RDERR: MOV R0,CSRA ;SAVE CSR ADDRESS
 MOV (R0),ACSR ;SAVE CONTENTS OF CSR
 JSP PC,ERR1 ;GO QUEUE RETURN FROM ERROR SERVICE
 MOV #1,ERRTYP ;ILLEGAL INTERRUPT OR DONE NOT SET
 ;*****
 HDRERS,REGIN,NULL ;ERROR RIT SET OR FALSE INTERRUPT
 ;*****
 JSP PC,ERR2 ;GO GET DEVICE NO. FROM ERROR RETURN QUE
 JMP R0RQ ;RETURN TO CALLER
 DATER: MOV R0,CSRA ;SAVE CSR ADDRESS
 TST (R0),AWAS ;GENERATE DRR ADDRESS IN R0
 MOV R1,AWAS
 MOV R2,DATTAR(R1),ASR
 MOV R3,PC,ASADR ;GO QUEUE RETURN FROM ERROR SERVICE
 JSP PC,ERR1
 MOV R1,ASADR
 MOV R2,(R0),(R1)
 TST -R0
 ;*****
 DATER,REGIN ;DATA ERROR!!!
 ;*****
 JSP PC,ERR2 ;GO GET DEVICE NO. FROM ERROR RETURN QUE
 HP ;RETURN TO READ ANOTHER FRAME
 RQ: .RLKW 9. ;16 BYTE FIFO QUEUE
 REQ: .RLKW 9. ;16 BYTE ERROR RETURN FIFO QUEUE
 ;SUPPORTINES TO QUEUE RETURN FROM ERROR SERVICE
 QERR1: MOV R1,REQP1 ;SAVE THE DEVICE NO IN THE QUEUE
 INC REQP1 ;UPDATE THE QUEUE POINTER
 CMP #REQ+20,REQP1 ;QUE POINTER AT HIGH LIMIT ??
 RPL 1S ;RR IF NOT
 MOV #REQ,REQP1 ;RESET THE POINTER
 1S: RTS PC ;RETURN TO REPORT THE ERROR
 QERR2: MOV #REQP2,R1 ;RETRIEVE THE DEVICE NO. FROM THE QUEUE
 INC REQP2 ;UPDATE THE QUE POINTER
 CMP #REQ+20,REQP2 ;POINTER AT THE HIGH LIMIT ??
 RNE 1S ;RR IF NOT
 MOV #REQ,REQP2 ;RESET THE POINTER

466 001410 010100
 467 001412 006300
 468 001412 006300
 469 001412 066700 176364
 470 001422 000207
 471
 472
 473
 474 001424 000000
 475 001424 000000
 476 001430 000000
 477 001430 000000
 478 001434 000000
 479 001436 000000
 480 001440 000000
 481 001442 000000
 482 001442 000000
 483 001444 000000
 484
 485 000001

1S: MOV R1,R0 ;DUPLICATE THE NO. IN R0
 ASL R0 ;GENERATE THE DEVICE REG OFFSET
 ASL R0
 ADD ADDR,R0 ;REBUILD THE CSR ADDRESS IN R0
 RTS PC
 ;SOME MODULE VARIABLES
 COUNT: OPEN ;COUNTER FOR 2048 INTR.
 COUNTC: OPEN ;COUNTER FOR READS ISSUED
 GETOUT: OPEN ;END PASS FLAG
 SYNFLC: OPEN ;DATA TABLE SYNC FLAG
 QPTR1: OPEN ;FIFO QUEUE POINTERS
 QPTR2: OPEN
 REQ1: OPEN ;ERROR RETURN QUEUE PTRS
 REQ2: OPEN
 FORK: OPEN ;RETURN POINTER FOR ERROR RETURNS
 ACTDEV: OPEN ;STORES NO. OF DEVICES ACTIVE
 .END

SPSIZ =	000040	1#	245		
SR1	000016R	205#			
SR2	000020R	206#			
SR3	000022R	207#			
SR4	000024R	209#			
SPART	000024R	211#	255#		
SAT	000026R	210#			
SUP	000012R	240#	334#		
SVRO	000052R	225#			
SVR1	000064R	226#			
SVR2	000066R	227#			
SVR3	000070R	229#			
SVR4	000072R	229#			
SVR5	000074R	230#			
SVR6	000076R	231#			
SYNPLG	001432R	335*	375	378*	477#
SYSCNT	000052R	225#			
IMPDPD=	000022	222#			
VECTOR	000010R	201#	272		
WASADR	000104R	235#	437*		
WDFR	000116R	242#	256*		
WDT0	000114R	241#			
XPLAG	000065R	199#			
.	001450R	291#	292#	449#	450#

. ABS. 000000 000
 001450 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

XPAAP0 XPAAP0/SOL/CRF:SYM=DDYCDW,XPAAP0
 RUN-TIME: 1 1 2 SECONDS
 RUN-TIME RATIO: 25.3=7.3
 CORE USED: 7K (14 PAGES)