

PAGE DEC/Y11 SYSTEM EXERCISER MODULE  
XPABFC.P11 12-OCT-78 12:04

MACV11 30A(1052) 12-OCT-78 16:55 PAGE 2

.REM -

IDENTIFICATION  
-----

PRODUCT CODE: AC-E857F-MC

PRODUCT NAME: CXP8F0 PA611 PUNCH MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

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

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

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

COPYRIGHT (C) 1973, 1978 DIGITAL EQUIPMENT CORPORATION

MAIN DEC CHANGE NOT  
MAY BE REQUIRED FOR  
PROGRAM TO OPERATE

- 1. ABSTRACT  
-----  
PAB IS AN IOMOD THAT EXERCISES UP TO 16 HIGH SPEED PUNCHES BY PUNCHING A STANDARD BINARY COUNT PATTERN ON EACH PUNCH. IF REPORTS ANY ERRORS ON THE CONSOLE THE MODULE IS DESIGNED TO ACTIVATE AND RUN ALL SELECTED DEVICES CONCURRENTLY.
- 2. REQUIREMENTS:  
-----  
HARDWARE: AT LEAST ONE PA-611-P CONTROL UNIT AND ONE PP67C/D PUNCH.  
STORAGE: PAB REQUIRES:  
1. DECIMAL WORDS: 376  
2. OCTAL WORDS: 6570  
3. OCTAL BYTES: 1360
- 3. PASS DEFINITION:  
-----  
ONE PASS OF THE PAB MODULE CONSISTS OF PUNCHING 2048. (TOTAL) CHARACTERS.
- 4. EXECUTION TIME:  
-----  
THE PAB RUNNING ALONE ON A PDP11/05 SYSTEM TAKES APPROXIMATELY --- MINUTES PER PASS.
- 5. CONFIGURATION REQUIREMENTS:  
-----  
DEFAULT PARAMETERS:  
DEVADP: 172700, VECTOR: 300, BR1:4, DEVCNT:1  
REQUIRED PAPAMETERS:  
AT CONFIGURATION TIME THE USER MUST MODIFY "VECTOR" IF ASSIGNED VALUE IS NOT 300.  
DEVICE/OPTION SET-UP:  
-----
- 6. A. INSURE ALL PUNCHES TO BE TESTED ARE LOADED WITH BLANK PAPER TAPE.  
B. TURN ON ALL PUNCHES TO BE TESTED.
- 7. MODULE OPERATION:  
-----

PABF DFC/X11 SYSTEM EXERCISER MODULE MACV11 3CA(1052) 12-OCT-78 16:55 PAGE 4  
XPABFC.P11 12-OCT-78 12:04

TEST SEQUENCE:

- A: CLEAR DEVICE AND DATA TABLES
- B: SET UP VECTORS FOR ALL ACTIVE DEVICES AND  
STORE DEVICE NO. FOR SELECTED DEVICES
- C: TURN ON INTERRUPT ENABLE FOR ALL ACTIVE DEVICES
- D: INTERRUPT SERVICE

- 1. TEST AND REPORT ANY ERROR BITS IN CSR
- 2. UPDATE AND OUTPUT NEXT CHARACTER IN PATTERN
- 3. WAIT FOR NEXT INTERRUPT
- 4. REPEAT 1 THROUGH 3 UNTIL 2096. (TOTAL)  
CHARACTERS HAVE BEEN PUNCHED

6. TURN OFF ALL ACTIVE DEVICES. REPORT END OF PASS  
AND RESTART AT C.

JSR TABLE:

TO LINK THE INDIVIDUAL INTERRUPTS WITH THE SERVICES. EACH THE SERVICE  
ROUTINE THERE IS A JSR TABLE CONTAINING 16 ENTRIES. WITHIN THE  
DEVICE VECTOR IS SET UP TO POINT TO A UNICORP SERVICE ROUTINE  
TABLE WHICH TRANSFERS CONTROL TO THE SERVICE ROUTINE USES TO  
AND POINTS R5 TO AN OFFSET THAT ADDRESS DATA TABLE ENTRY.  
GENERATE THE CORRECT REGISTER ADDRESS AND THEN EXECUTE A THE

FIFO QUEUE:

TO ALLOW THE SERVICE ROUTINE TO USE THE SAME GPR'S FOR  
SERVICING UP TO 16 CONCURRENT INTERRUPTS. THIS IS ACCOMPLISHED BY  
STORING A FIFO (FIRST-IN-FIRST-OUT) QUEUE. THE OFFSET IN THE  
ENTRIES AND THE INTERRUPT SERVICE AND THEN EXECUTE A THE  
QUEUE UPDATES THE QUEUE POINTER A LOWER PRIORITY THAT  
CONTINUE SERVICING THE QUEUE WHEN A LOWER PRIORITY THAT  
INTERRUPTS REQUESTS.

ERROR RETURN QUEUES:

WHEN AN "ERROR" OR "DATA" ADDRESS IS EXECUTED, THE MONITOR  
QUEUES UP THE RETURN ADDRESS TO HAVE BEEN QUEUED UP SINCE IT IS  
POSSIBLE FOR ANOTHER LINE CONTROL TO BE QUEUED UP AT A HIGHER  
PRIORITY. IT AND CLOSURE WHEN THE REGISTER CONTROL AGAIN AFTER  
SERVICE ROUTINES LINE TO PREVENT THIS FROM HAPPENING (LINE OFFSET)  
BY THE FIFO REGISTER LINE TO PREVENT THIS FROM HAPPENING (LINE OFFSET)  
THE ERROR TO PREVENT THIS FROM HAPPENING (LINE OFFSET)  
IS SAVED IN AN ERROR RETURN QUEUE. THE RETURN ADDRESS IS  
AND RESTORED FROM THE QUEUE WHEN A 16 BYTE QUEUE (FIFO) REQ-  
UIRED TO ACCOMPLISH THIS TASK:

TEQ XMTR ERROR RETURN QUEUE

TO CONTROL THE OPERATION OF THE QUEUE TWO POINTERS ARE  
REQUIRED:

TEQP1 SAVES R1 IN THE XMTR QUEUE  
TEQP2 RETRIEVES R1 FROM THE XMTR QUEUE

OPERATION OPTIONS:

-----  
A. LOCATION DVID1 (PAB14) MAY BE CHANGED TO SELECT ANY  
COMBINATION OF DEVICES TO BE TESTED. BIT0=DEV0  
BIT1=DEV1 ..... BIT15=DEV15 IF DVID1=0 THE MODULE WILL BE  
DROPPED FROM TEST.

B. MEANING OF SRI BITS 7 THRU 0  
SET = DISABLE PUNCHING ON THAT DATA CHANNEL.  
CLEAR = ALLOW

NON-STANDARD PRINTOUTS  
-----

9.

NONE: ALL PRINTOUTS HAVE THE SAME FORMATS DESCRIBED  
IN THE DEC/X11 DOCUMENT.

PABF11P DEC/X11 EXERCISER MODULE
G000000 IGWDD <PARF > 172700 0 4 2048 54
0000000 140000 PABF 172700 0 4 2048 54
TITLE PABF DEC/X11 SYSTEM EXERCISER MODULE
DDXCO# VERSION 6 23-MAY-78

\*\*\*\*\*
BEGIN:
MODNAM: .ASCII /PARF / MODULE NAME
XFLAG: .RYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR: 172700+0 ;ACT DEVICE ADDR
VECTOR: C+A ;ACT DEVICE VECTOR.
BRI: .BYTE PRTV4+0 ;1ST RR LEVEL.
PR2: .BYTE PRTV+0 ;2ND RR LEVEL.
SBI: +1 ;SWITCH REGISTER 1.
SR1: OPEN ;SWITCH REGISTER 2
SR2: OPEN ;SWITCH REGISTER 3
SR3: OPEN ;SWITCH REGISTER 4
SR4: OPEN ;SWITCH REGISTER 5
\*\*\*\*\*
STAT: 140000 ;STATUS WORD
INIT: STAPT ;MODULE START ADDR.
MODSP: MODSP ;MODULE STACK POINTER.
PASCNT: C ;PASS COUNTER.
ICDWT: C C48. ;LOC TO COUNT ITERATIONS
ICOUNT: C ;LOC TO SAVE TOTAL HARD ERRORS
SOPCNT: C ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT: C ;LOC TO SAVE SOFT ERRORS PER PASS
SOPPAS: C ;LOC TO SAVE HARD ERRORS PER PASS
HRDPAS: C ;% OF SVS ERRORS ACCUMULATED
SYSCNT: C ;HOLDS PANDOW # WHEN RAND MACRO IS CALLED
RANNUM: C ;RESERVED FOR MONITOR USE
CONFIG: C ;RESERVED FOR MONITOR USE
RES1: C ;RESERVED FOR MONITOR USE
SVR0: OPEN ;LOC TO SAVE R0
SVR1: OPEN ;LOC TO SAVE R1.
SVR2: OPEN ;LOC TO SAVE R2.
SVR3: OPEN ;LOC TO SAVE R3.
SVR4: OPEN ;LOC TO SAVE R4.
SVR5: OPEN ;LOC TO SAVE R5.
SVR6: OPEN ;LOC TO SAVE R6.
CSRA: OPEN ;ADDR OF CURRENT CSR.
SRADR: ;ADDR OF GOOD DATA, OR
ASADR: OPEN ;CONTENTS OF CSR.
WASADR: ;ADDR OF BAD DATA, OR
ASTAT: OPEN ;STATUS REG CONTENTS.
ERRTYP: ;TYPE OF ERROR
ASR: OPEN ;EXPECTED DATA.
AWAS: OPEN ;ACTUAL DATA.
RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
WTO: OPEN ;WORDS TO MEMORY PER ITERATION
WFP: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;% OF INTERRUPTS PER ITERATION

IDNUM: 54 ;MODULE IDENTIFICATION NUMBER=54
DEPT SPSIZ ;MODULE STACK STARTS HERE.
WORD C
LIST
ENDR
MODSP:
\*\*\*\*\*
;MODULE INITIALIZATION ROUTINES
START: MOV #1024, WETO ;1024 WORDS TO MEM
MOV #2048, INTR ;2048. INTERRUPTS
JSP PC, STRT1 ;GO INITIALIZE MODULE VARIABLES
;ROUTINE TO CLEAR DEVICE AND DATA TABLES (32 BYTES TOTAL)
MOV #DEVTAB, PC ;INITIALIZE TABLE POINTER
CLR (R0)
CMP R0, #DEVTAB+40
BNE 1S ;RR IF NOT FINISHED 32 BYTES
;ROUTINE TO DETERMINE DEVICES TO BE TESTED AND SET UP VECTORS
STI: MOV DVID1, R0 ;GET DEVICE SELECTION PARAMETERS
BNE 1S ;RR IF ANY SELECTED FOR TEST
ENBRS, REG IN
1S: CLR R1 ;SET UP R1, R0 TO START AT ZERO
CLR R1
CLR R2
CLR R3
CLR R4
CLR R5
CLR R6
MOV VECTOR, R3 ;START AT BEGINNING OF VECTOR AREA
MOV JSRTAB, R4 ;START AT BEGINNING OF JSRTABLE
COUNT, R5 ;COUNT THE DEVS
ASR PC ;SHIFT SELECT BIT INTO "C"
BCS 4S ;GO SET IT UP
ADD #4, R4 ;UPDATE THE VECTOR AND JSRTAB PTRS.
CMP #19, R1 ;LAST POSSIBLE DEVICE?
BNE 4S ;RR IF NOT
MOV R4, (R3)+ ;GO START UP ALL ACTIVE DEVICES
MOV R1, (R3)+ ;LOAD TINTR POINTER IN VECTOR
INC R2 ;LOAD PRIORITY LEVEL IN VECTOR
MOV R1, DEVTAB(R2) ;COUNT ACTIVE DEVICES
BR 3S ;STORE ACTIVE DEVICE NO. IN
MOV SR1, DATMSK ;GO CHECK FOR END
BR RSTRT ;SET UP PUNCH MASK
RSTRT: ;ROUTINE TO INITIALLY START UP ALL ACTIVE DEVICES
MOV R2, ACTDEV ;SAVE NO. OF ACTIVE DEVICES
MOV R1, DEVTAB(R2), R0 ;GET ACTIVE DEVICES NO.
ASL R0 ;GENERATE REG. OFFSET
ADD R0, ADDR, R0 ;GEN. CSR ADDRESS

292 000406 012710 000100  
293 000412 005367 000662  
294 000416 005302  
295 000420 104364  
296 000422 104400 000000  
297  
298  
299  
300 000426 000010  
301 000446 000010  
302  
303  
304  
305  
306 000455 004567 000134  
307 000472 000000  
308 000474 004567 000126  
309 000500 000004  
310 000502 004567 000120  
311 000504 000010  
312 000510 004567 000112  
313 000514 000014  
314 000516 004567 000104  
315 000524 000220  
316 000524 000076  
317 000530 000324  
318 000532 004567 000070  
319 000536 000030  
320 000540 004567 000062  
321 000540 000034  
322 000544 004567 000054  
323 000552 000140  
324 000554 004567 000046  
325 000562 000044  
326 000562 000040  
327 000566 000150  
328 000570 004567 000032  
329 000574 000054  
330 000575 004567 000024  
331 000605 000030  
332 000604 004567 000016  
333 000610 000064  
334 000612 004567 000010  
335 000620 004567 000002  
336 000624 000074  
337  
338  
339  
340  
341 000626 111577 000450  
342 000632 005257 000444  
343 000636 026227 000440 001340  
344 000644 012003  
345 000646 012003 001320 000426  
346 000654 012005  
347

MOV #100,(R0) ;TURN ON INTERRUPT ENAB.  
DEC COUNTC ;KEEP TRACK OF HOW MANY PUNCHES  
DEC R2  
BPL 16  
EXITS,BEGIN ;BR UNTIL ALL ARE STARTED  
;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.  
  
;ACTIVE DEVICE AND DATA TABLES  
DEVTAB: .BLKW 8: ;16 BYTE ACTIVE DEVICE TABLE  
DATTAB: .BLKW 8: ;16 BYTE DATA TABLE  
  
;JSR TABLE TO PROVIDE LINKAGE BETWEEN DEVICE NO. AND INTR. SERVICE  
JSRTAB: JSR R5,TINTR ;LINK TINTR TO DEV. 0  
JSR R5,TINTR ;LINK TINTR TO DEV 1  
JSR R5,TINTR ;LINK TINTR TO DEV 2  
JSR R5,TINTR ;LINK TINTR TO DEV 3  
JSR R5,TINTR ;LINK TINTR TO DEV 4  
JSR R5,TINTR ;LINK TINTR TO DEV 5  
JSR R5,TINTR ;LINK TINTR TO DEV 6  
JSR R5,TINTR ;LINK TINTR TO DEV 7  
JSR R5,TINTR ;LINK TINTR TO DEV 10  
JSR R5,TINTR ;LINK TINTR TO DEV 11  
JSR R5,TINTR ;LINK TINTR TO DEV 12  
JSR R5,TINTR ;LINK TINTR TO DEV 13  
JSR R5,TINTR ;LINK TINTR TO DEV 14  
JSR R5,TINTR ;LINK TINTR TO DEV 15  
JSR R5,TINTR ;LINK TINTR TO DEV 16  
JSR R5,TINTR ;LINK TINTR TO DEV 17  
  
;ALL TRANSMITTER INTERRUPTS ENTER HERE VIA JSR TABLE  
TINTR: (P5),QPTR1 ;STORE REQUEST IN FIFO QUEUE  
INC QPTR1 ;UPDATE THE QUEUE POINTER  
CMP QPTR1,#TQ+20 ;POINTER BEYOND LIMIT  
BNE 15 ;BR IF NOT  
MOV #TQ,QPTR1 ;RESET Q POINTER  
MOV (SP)+,R5 ;PUT BACK OTHER GUY'S R5  
15: ;-----

348 000656 000004 000000 000664  
349  
350  
351  
352  
353 000664 005767 000424  
354 000670 104464  
355 000672 117700 000406  
356 000678 110001  
357 000700 005267 000400  
358 000700 005267 000400 001340  
359 000714 005123  
360 000714 012767 001320 000362  
361 000722 005700 177000  
362 000722 005367 000344  
363 000722 005367  
364 000734 005167 000354  
365 000740 000077  
366 000742 104413 000000  
367  
368  
369 000746 005710  
370 000750 100004  
371 000752 112767 000762 000337  
372 000760 005432  
373 000762 005710  
374 000764 100404  
375 000765 112767 000776 000316  
376 000774 000424  
377 000776 005701  
378 001000 006201  
379 001002 005767 000272  
380 001006 001415  
381 001010 005367 000264  
382 001010 105261 000446  
383 001020 005167 000446 000246  
384 001025 116767 000243 000246  
385 001034 116760 000234 000002  
386 001042  
387 001042 104400 000000  
388  
389 001046 011967 177030  
390 001052 010067 177022  
391 001056 110177 000224  
392 001066 022767 001360 000212  
393 001074 001006  
394 001076 012767 001340 000202  
395 001104 012767 000011 176774  
396 001112  
397  
398  
399 001112 104405 000000 000000  
400  
401 001120 117701 000164  
402 001120 005267 000160  
403 001130 022767 001360 000152

PIPO5,BEGIN,TSERV ; QUEUE UP TO CONTINUE AT TSERV AND RTI  
;-----  
;ROUTINES TO SERVICE INTR REQUEST QUEUE  
TSERV: TST GETOUT ;END PASS FLAG SET ??  
BPL TSER1 ;BR IF YES  
MOVR #QPTR2,R0 ;GET DEVICE REG. OFFSET  
MOV R0,R1 ;DUPLICATE OFFSET IN R1  
INC QPTR2 ;UPDATE THE QUEUE POINTER  
CMP QPTR2,#TQ+20 ;TEST LIMIT  
BNE 15 ;BR IF NOT BEYOND LIMIT  
MOV #TQ,QPTR2 ;RESET Q POINTER  
ADD ADDR,R0 ;GENERATE CSR ADDRESS  
DFC COUNT ;COUNT 2048. INTR.  
BNE 2S ;BR IF NOT  
BP GETOUT ;SET END PASS FLAG  
 ;BR IF 2048. INTR  
2S:  
ENDITS,BEGIN ;SIGNAL END OF ITERATION.  
TST (P0) ;MONITOR SHALL TEST END OF PASS  
BPL TERO ;ERROR FLAG SET IN CSR ??  
MOV #TEPG,FORK ;SET UP RETURN TO TERO  
BR TERO ;GO REPORT THE ERROR  
TSTR (P0) ;WAS DONE SET ??  
BWI TEP1 ;BR IF SET  
MOV #TER1,FORK ;SET UP RETURN TO TER1  
BR TERO ;GO REPORT THE ERROR  
TER1: ASR R1 ;GEN. DATA TABLE INDEX  
ASR R1  
TST COUNTC ;HAVE WE PUNCHED ENOUGH?  
BEQ TSER1 ;BR IF YES  
DEC COUNTC ;OTHERWISE DO ANOTHER  
INCR DATTAB(R1) ;BUMP DATA CHAR +1  
MOVR DATTAB(R1),DATHLD ;LOAD XMIT DBR  
JSR DATMSR,DATHLD ;ELIMINATE UNWANTED BITS  
MOVR DATTABLD,2(R0) ;NOW LOAD XMIT DBR  
TSEP1:  
EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.  
TERR: MOV (R0),ACSR ;SAVE CONTENTS OF CSR  
MOV R0,CSRA ;SAVE CSR ADDRESS  
MOVR R1,TEQP1 ;SAVE THE DEVICE OFFSET IN THE QUEUE  
INC TEQP1 ;UPDATE THE QUEUE POINTER  
CMP #TEQ+20,TEQP1 ;POINTER AT THE HIGH LIMIT ??  
BNE 15 ;BR IF NOT  
MOV #TEQ,TEQP1 ;RESET THE POINTER  
MOV #11,ERRTYP ;DONE NOT SET OR ILLEGAL INTERRUPT  
15:  
;\*\*\*\*\*  
;ORDERS,BEGIN,NULL ;ERROR BIT OR FALSE INTR.  
;\*\*\*\*\*  
MOVR #TEQP2,R1 ;RETRIEVE THE DEVICE NO. FROM THE QUE  
INC #TEQ2 ;UPDATE THE QUEUE POINTER  
CMP #TEQ+20,TEQP2 ;POINTER AT THE HIGH LIMIT ??

```

404 001136 001003 001340 000142 BNE 2S JBR IF NOT
405 001140 012767 001340 000142 MOV #TEQ,TEQP2 JRESET THE QUEUE POINTER
406 001150 066700 176632 000142 ADD ADDR,R0 JDUPLICATE THE OFFSET IN R0
408 001154 000177 000132 JMP @FORK JRETURN THE CSR ADDRESS
409 JEND OF PASS ROUTINE TO DISABLE ALL ACTIVE DEVICES
410
411 001160 016701 000132 ENPS: MOV ACTDEV,R1 JGET NO. OF ACTIVE DEVICES
412 001164 116100 00042A IS: MOV DEVTAR(R1),R0 JGET A DEVICE NO. FROM DEV. TABLE
413 001170 006300 ASL R0 JGENERATE CSR ADDRESS
414 001174 006300 ASL R0
415 001174 066700 ADD ADDR,R0
416 001200 005010 CLR (R0) JTURN OFF ONE ACTIVE READER
417 001202 0053C1 DEC R1 JCOUNT IT
418 001204 102367 BPL IS JBR TIL THEY ARE ALL OFF
419 001206 004767 JSR PC,STRT1 JGO REINITIALIZE THE MODULE VARIABLES
421 001212 104413 000004 ENDS,BEGIN JSIGNAL END OF ITERATION.
422 000000 JMONITOR SHALL TEST END OF PASS
423
424 001216 016767 176614 000052 STRT1: MOV ICONT,COUNT JSET UP FOR 2048 INTR. PER PASS
425 001222 012767 004000 000046 MOV #2048,COUNTC JSET UP FOR 2048 PUNCHES
426 001224 012767 001320 000042 MOV #0,OPTR1 JSET UP FIFO SERVICE QUEUE POINTERS
427 001240 012767 001320 000036 MOV #0,OPTR2
428 001246 012767 001340 000032 MOV #TEQ,TEQP1 JSET UP THE ERROR RETURN QUE POINTERS
429 001252 012767 001340 000032 MOV #TEQ,TEQP2
430 001262 005767 000032 CLR GETOUT
431 001266 016702 000024 MOV ACTDEV,R2 JINITIALIZE END OF PASS FLAG
432 001272 0002C7 RTS PC JLOAD R2 WITH NO. OF ACTIVE DEVICES
433 JRETURN TO CALLING ROUTINE
434
435
436 JSOME MODULE VARIABLES
437 001274 000 DATHLD: .BYTE OPEN JTMP HLDING FOR MASKING
438 001275 000 DATMSK: .BYTE OPEN JMASK TO CHOP OFF BITS
439 001276 000 COUNT: OPEN JCOUNTS NO. OF INTR. PER PASS
440 001300 000 COUNTC: OPEN JCOUNTS NO. OF PUNCHES ISSUED
441 001302 000 OPTR1: OPEN JFIFO QUEUE POINTERS
442 001304 000 OPTR2: OPEN
443 001306 000 TEQP1: OPEN JERROR RETURN QUE POINTERS
444 001308 000 TEQP2: OPEN
445 001312 000 FORK: OPEN JERROR RETURN POINTER
446 001314 000 GETOUT: OPEN JENDPASS FLAG
447 001316 000 ACTDEV: OPEN JCONTAINS NO. OF ACTIVE DEVICES
448 001320 000 TQ: .RFLW 8: J16 BYTE FIFO QUEUE
449 001340 000 J16 BYTE ERROR RETURN QUEUE
450
451 0000C1 .END

```

```

ACSR 000102R 226# 399*
ACTDEV 001316R 244#
ADDR 000000R 102#
ADDR22= 001000 244#
ASB 000106R 230#
ASRT 000104R 229#
AWAST 000110R 244#
BEGIN 000000R 100# 263 296 348 367 387 399 421
BIT0 = 000001 244#
BIT1 = 000002 244#
BIT10 = 004000 244#
BIT11 = 004000 244#
BIT12 = 010000 244#
BIT13 = 020000 244#
BIT14 = 040000 244#
BIT15 = 100000 244#
BIT16 = 000000 244#
BIT3 = 000010 244#
BIT4 = 000020 244#
BIT5 = 000040 244#
BIT6 = 000080 244#
BIT7 = 000100 244#
BIT8 = 000200 244#
BIT9 = 000400 244#
BREAK = 001000 244#
BR1 = 104401R 194#
BR2 = 000013R 105# 279
BTODS = 104411 244#
COATA = 104412 244#
COMPTC 000056R 214#
COUNT 001276R 304# 424*
COUNTC 001308R 379# 425* 440*
CSRA 000100R 224# 390*
DATCK = 104411 244#
DATEPR = 104404 244#
DATEPRR 001274R 303# 384*
DATTAB 000446R 301# 385# 437*
DEVTAB 000426R 253# 382* 383#
DVID1 = 000014R 263# 255# 288 300# 413
ENDITS = 104413 196# 421
ERRRTP 000106R 229# 305#
ERRRTPR 001312R 303# 375#
GETPA = 104415 244# 387#
GETPA = 104414 244# 408# 445#
GETPA = 104414 244# 430# 446#
HRTPT = 000044R 209#
HRTPT = 000044R 244#
HRTPT = 000044R 244#
ICONT 000038R 208# 399
ICONT 000040R 207#
IDONW 000122R 236#
INIT 000030R 203# 424

```





DIAGNOSTIC ENGINEERING



DECO  DEPO  SUBMISSION

FOR RELEASE ENG. USE  
 NEW  CHANGE  DELETE

PRODUCT IDENTIFICATION													
MD	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE			STATUS	DISTRIBUTION	1ST COPY - RIGHT YEAR	LAST COPY - RIGHT YEAR	
						DD	MMM	YY					
	ZZ	CXPAB	F	1	01	3	APR	79	OBSOLETE	X G	R	1973	1979

TITLE: CXPAB/ PA611 PUNCH MODULE

AUTHOR: D. BUTENHOF      MAINTAINING GROUP: DEC/X11 SPT GP      MAINTAINER: D. BUTENHOF      SUBMITTING ENGINEER: D. BUTENHOF

PRODUCT COMPONENTS							
CK	DESCRIPTION	PRODUCT NO.	REV	CK	DESCRIPTION	PRODUCT NO.	REV
	DOCUMENT				INDEX		
	LISTING				SOURCE MEDIA		
	OBJECT MEDIA				TEST MEDIA		
X	DECO	AF-E857F-M1					

PRODUCTS OBSOLETE (other than previous version)											
LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV
MD			MD			MD					

PROCESSORS PRODUCT OPERATES WITH (Enter all applicable 2-digit codes representing the Processor the product operates with. See separate instructions.)

OPERATIONAL CODES (Enter all applicable 2-digit codes that describe the product. See separate instructions.)

ACT/APT/XXDP	EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME
INFORMATION FIELD			<input type="checkbox"/> Y <input type="checkbox"/> N	<input type="checkbox"/> Y <input type="checkbox"/> N	SECONDS	SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED: \_\_\_\_\_

DEVICE AFFECTED: DEC/X11      MULTIMEDIA AFFECTED?  YES  NO

KIT NUMBERS	ZJ129-RZ, FR	ZJ215-FR	ZJ239-RZ, PB	ZJ240-RB, RE	ZJ240-FR
	ZJ215-RY, RZ	ZJ239-RB, RY	ZJ239-FR	ZJ240-RZ, PB	

PROBLEM: Unable to handle multiple devices correctly

SOLUTION: WITH THE FOLLOWING PATCH, WILL HANDLE ONLY ONE DEVICE

DEPO PATCH AREA					
CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO
366	10267	12700	420	100	240
370	724	172700			
372	116200	5010			
374	426	240			
376	6300	240			
400	6300	240			
402	66700	240			
404	177400	240			

SUBMITTING ENGINEER: <i>D. Butenhof</i>	MANUFACTURING ENGINEER: <i>John P. Butch</i>	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER
DATE: 3 APR 79	DATE: 25 APR 79	DATE:	098-05460
MAINTAINER: <i>D. Butenhof</i>	FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. 3064
DATE:	DATE:	DATE:	