

IDENTIFICATION  
-----

PRODUCT CODE: AC-E893R-MC  
PRODUCT NAME: CXDPBB0 DUP-11 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) 1975,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

DPB IS AN IOMOD THAT EXERCISES UP TO EIGHT CONSECUTIVELY  
ADDRESSED SYNCHRONOUS LINE INTERFACES (DUPI) BY TRANSMITTING A STANDARD  
BINARY COUNT PATTERN USING THE SYSTEM TEST MODE FEATURE. THE  
RECEIVED DATA IS COMPARED WITH THE TRANSMITTED DATA AND ANY ERRORS  
ARE REPORTED VIA THE CONSOLE ITTY. ALL AVAILABLE INTERFACES  
(UP TO 8) ARE ACTIVATED AND RUNNING SIMULTANEOUSLY.

2. REQUIREMENTS

HARDWARE: DUP11 SYNCHRONOUS INTERFACE  
STORAGE: DPB REQUIRES:

1. DECIMAL WORDS: 585
2. OCTAL WORDS: 1111
3. OCTAL BYTES: 2222

3. PASS DEFINITION

ONE PASS OF THE DPB MODULE CONSISTS OF TRANSMITTING AND RECEIVING  
48,080 8-BIT CHARACTERS (TOTAL) PER ACTIVE DEVICE.

4. EXECUTION TIME

DPB RUNNING ALONE ON A PDP11/40 PROCESSOR TAKES APPROXIMATELY  
1 MINUTE TO COMPLETE ONE PASS. THIS TIME INCREASES SLIGHTLY  
AS MORE DEVICES ARE ADDED.

5. CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DEVADR: 000001 VECTOR: 001 BR1:5 BR2:5 DEVCNT:1  
THE DEVICE IS TESTED IN SDLC MODE AS A SECONDARY STATION,  
WITH A HARDWARE BCC CHECK USING THE CRC/CCITT POLYNOMIAL.

REQUIRED PARAMETERS: THE CSR AND VECTOR MUST BE SET UP BY THE USER.

6. DEVICE/OPTION SETUP

NONE: NO SPECIAL SETUP IS REQUIRED IN SYSTEM TEST MODE

7. MODULE OPERATION  
-----

TEST SEQUENCE:

- A. TEST UP TO 8 POSSIBLE DEVICES FOR SELECTION
- B. STORE THE NO. OF DEVICES TO BE TESTED AND SET UP THE SECTORS AND PRIORITIES FOR THESE DEVICES
- C. LOAD ALL REGISTERS--SECONDARY STATION RECEIVER AND TRANSMITTER ON, SYSTEM TEST MODE AND INTERRUPT ENABLES FOR ALL ACTIVE DEVICES. ENABLE SELECTED DEVICES.
- D. TRANSMITTER INTERRUPT SERVICE:
  - 1.) TEST FOR FALSE INTERRUPT (READY (0)); REPORT ERRORS
  - 2.) OUTPUT NEXT CHARACTER TO THE DEVICE
  - 3.) RETURN TO MONITOR TO WAIT FOR RECEIVER INTERRUPT.
- E. RECEIVER INTERRUPT SERVICE:
  - 1.) TEST FOR FALSE INTERRUPT (DONE (0)); REPORT ERRORS
  - 2.) CHECK FOR DATA ERROR; WAIT FOR TRANSMITTER INTERRUPT
  - 3.) RETURN TO MONITOR TO WAIT FOR TRANSMITTER INTERRUPT
- F. REPEAT D AND E UNTIL ALL DEVICES HAVE BEEN PROCESSED THEN CHECK FOR BCC ERRORS
- G. TURN OFF ALL ACTIVE DEVICES AND DECREMENT ITERATION COUNT. IF NOT 0, RESTART AT B.
- H. SIGNAL END PASS.

8. OPERATION OPTIONS  
-----

- A. LOCATION DVID1 (DPB 14) MAY BE CHANGED TO SELECT ANY COMBINATION OF DEVICES BIT0=DEVO, BIT1=DEV1 .....BIT7=DEV7.  
NOTE: IF DVID1 IS INITIALLY 0 DPB WILL BE DROPPED FROM TEST.

9. NON STANDARD PRINTOUTS  
-----

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

```

141 000000
142 000000
143 000000
144 000000
145 000000
146 000000
147 000000
148 000000
149 000000 050104 041102 040
150 000005 000 000
151 000006 000001 000
152 000010 000001 000
153 000012 240 240
154 000013 240 240
155 000014 000001 000
156 000016 000000 000
157 000020 000000 000
158 000022 000000 000
159 000024 000000 000
160 000026 140000 140000
161 000030 000240 000240
162 000033 000224 000224
163 000034 000000 000000
164 000036 000000 000000
165 000036 000040 000040
166 000040 000000 000000
167 000042 000000 000000
168 000042 000000 000000
169 000046 000000 000000
170 000050 000000 000000
171 000054 000000 000000
172 000054 000000 000000
173 000056 000000 000000
174 000056 000000 000000
175 000060 000000 000000
176 000062 000000 000000
177 000064 000000 000000
178 000066 000000 000000
179 000070 000000 000000
180 000072 000000 000000
181 000074 000000 000000
182 000076 000000 000000
183 000100 000000 000000
184 000102 000000 000000
185 000102 000000 000000
186 000104 000000 000000
187 000104 000000 000000
188 000106 000000 000000
189 000106 000000 000000
190 000110 000000 000000
191 000112 000254 000254
192 000114 000000 000000
193 000116 000000 000000
194 000120 000000 000000
195 000122 000070 000070

```

```

IOMOD <DPBB> 1,1,5,5,0,40,70
MODULE 140000,DPBB,1,1,5,5,0,40,70
TITLE DPBB DEC/X11 SYSTEM EXERCISER MODULE
DDICOM VERSION 6 23-MAY-78
LIST BIN
*****
BCIN:
MODNAM: .ASCII /DPBB / ;MODULE NAME.
XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WRUFF USAGE
ADDR: 1=0 ;1ST DEVICE ADDR.
VECTOR: 1=0 ;1ST BR LEVEL.
BR1: .BYTE PRTV5+0 ;1ST BR LEVEL.
BR2: .BYTE PRTV5+0 ;2ND BR LEVEL.
DID1: 0=1 ;DEVICE INDICATOR 1.
SR1: OPEN ;SWITCH REGISTER 1
SR2: OPEN ;SWITCH REGISTER 2
SR3: OPEN ;SWITCH REGISTER 3
SR4: OPEN ;SWITCH REGISTER 4
*****
STAT: 140000 ;STATUS WORD
INIT: START ;MODULE START ADDR.
SPDINT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER.
ICOUNT: 40 ;# OF ITERATIONS PER PASS=40
ICOUNT: 0 ;LOC TO COUNT ITERATIONS
SOPCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCHT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
SRPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
SVSCHT: 0 ;# OF SYS ERRORS ACCUMULATED
RANRND: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
CNFVIC: 0 ;RESERVED FOR MONITOR USE
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;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.
SADR: OPEN ;ADDR OF GOOD DATA, OR
ASDR: OPEN ;CONTENTS OF CSR.
WASADR: OPEN ;ADDR OF BAD DATA, OR
ASTAT: OPEN ;STATUS REG CONTENTS.
ERRTYP: OPEN ;TYPE OF ERROR
ASB: OPEN ;EXPECTED DATA.
AMAS: OPEN ;ACTUAL DATA.
RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: 70 ;MODULE IDENTIFICATION NUMBER=70

```

```

196 000040
197 000040
198 000040
199 000040
200 000224
201 000224
202 000224

```

```

-REPT SPSIZ ;MODULE STACK STARTS HERE.
-WORD 0
-LIST
-ENDR
MODSP:
*****

```

```

203
204
205 100000
206 040000
207 000000
208 010000
209 004000
210 002000
211 001000
212 000400
213 000200
214 000100
215 000040
216 000020
217 000010
218 000004
219 000002
220 000001
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
256
257
258

```

```

;BIT DEFINITIONS
BIT15=100000
BIT14=40000
BIT13=000000
BIT12=010000
BIT11=004000
BIT10=002000
BIT9=001000
BIT8=000400
BIT7=000200
BIT6=000100
BIT5=000040
BIT4=000020
BIT3=000010
BIT2=000004
BIT1=000002
BIT0=000001

```

```

;CONTROL REGISTER DEFINITIONS
-----
;RXCSR BIT DEFINITIONS
DSCA=BIT15 ;DATA SET CHANGE A
RINC=BIT14 ;PING
CSS=BIT14 ;CLR TO SEND
CARDET=BIT12 ;CARRIER DETECT
RECACT=BIT11 ;REC ACTIVE
SRD=BIT10 ;SEC REC DATA
DSR=BIT9 ;DATA SET RDY
STPSYN=BIT8 ;STRIP SYNC
RXDONE=BIT7 ;REC DONE
RINFE=BIT6 ;REC INTR ENABLE
DSINFE=BIT5 ;DSC INTR ENABLE
RCVEN=BIT4 ;REC ENABLE
STD=BIT3 ;SEC XMIT DATA
REQ=BIT2 ;REQ TO SEND
DTR=BIT1 ;REC START OF MESSAGE
DSCB=BIT0 ;DATA SET CHANGE B

```

```

;RXDBUF BIT DEFINITIONS
RXDERR=BIT15 ;REC DATA ERROR
OVRERR=BIT14 ;OVERRUN ERROR
CRCERR=BIT12 ;CRC ERROR
RABORT=BIT10 ;REC ABORT
RDOM=BIT9 ;REC END OF MESSAGE
RSON=BIT8 ;REC START OF MESSAGE

```

```

;PARCSR BIT DEFINITIONS
DECMOD=BIT15 ;DEC MODE (DDCMP)
CRCEN=BIT9 ;CRC ENABLE
PRISEC=BIT12 ;PRIS/SEC SELECT

```

```

;TXCSR BIT DEFINITIONS
TXDLAT=BIT15 ;TX DATA LATE
MDATA=BIT14 ;MAINT DATA OUT
CLK=BIT1 ;CLK
MMODEB=BIT12 ;MAINT MODE B
MMODEA=BIT11 ;MAINT MODE A

```

```

259 002000
260 001000
261 000400
262 000200
263 000100
264 000020
265 000010
266
267 000000
268 014000
269 010000
270 004000
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312

```

```

BITW=BIT10 ;BIT WINDOW INPUT
TXACT=BIT9 ;TX ACTIVE
MRESET=BIT8 ;MAINT RESET
TXDONE=BIT7 ;XMIT DONE
TXINFE=BIT6 ;TX INTR ENABLE
SEMD=BIT4 ;SEND
HDXEN=BIT3 ;HD/FOX

```

```

;TXCSR WRD DEFINITIONS
USER=0 ;USER MODE
MMODE=14000 ;MAINT INT MODE
MEXT=10000 ;MAINT EXT MODE
SYSTST=4000 ;SYSTEM TEST MODE

```

```

;TXDBUF BIT DEFINITIONS
RCRC7T=BIT15 ;CRC BIT WINDOW
RCRCIN=BIT14 ;DITTO
TCRC7T=BIT13 ;DITTO
TCRCIN=BIT12 ;DITTO
TIMER=BIT11 ;MAINTENANCE TIMER
TABORT=BIT10 ;TRANSMIT ABORT
TEOM=BIT9 ;TRANSMIT END OF MESSAGE
TSON=BIT8 ;TRANSMIT START OF MESSAGE

```

```

;MISC. PROGRAM DEFINITIONS AND EQUATES
RXCSR=R0 ;RECEIVER CONTROL REGISTER
RXDBUF=R7 ;RECEIVER DATA BUFFER REGISTER
PARCSR=R ;PARAMETER STATUS REGISTER
TXCSR=R4 ;TRANSMITTER CONTROL REGISTER
TXDBUF=R6 ;TRANSMITTER DATA BUFFER REGISTER

```

```

N.DUPS: OPEN ;SOFTWARE MEMORY
TOTAL: OPEN ;STORAGE FOR DATA CHECK
COUNT: OPEN ;ITERATION COUNT
RMB: OPEN ;REC BUFFER ADRS STORAGE
SAVBF: OPEN ;REC ISR TEMPORARY STORAGE
SELECT: OPEN ;SOFTWARE POINTER TO ACTIVE DEVICES

```

```

-MACRO SPUSH SA,SB,SC,SD,SE,SF
-IF NB <<SA>> MOV SA,-(SP) ;SAVE SA ON THE STACK
-IF NB <<SB>> MOV SB,-(SP) ;SAVE SB ON THE STACK
-IF NB <<SC>> MOV SC,-(SP) ;SAVE SC ON THE STACK
-IF NB <<SD>> MOV SD,-(SP) ;SAVE SD ON THE STACK
-IF NB <<SE>> MOV SE,-(SP) ;SAVE SE ON THE STACK
-IF NB <<SF>> MOV SF,-(SP) ;SAVE SF ON THE STACK
-ENDM
-MACRO SPOP SA,SB,SC,SD,SE,SF
-IF NB <<SA>> MOV (SP),SA ;POP STACK TO SA
-IF NB <<SB>> MOV (SP),SB ;POP STACK TO SB
-IF NB <<SC>> MOV (SP),SC ;POP STACK TO SC
-IF NB <<SD>> MOV (SP),SD ;POP STACK TO SD
-IF NB <<SE>> MOV (SP),SE ;POP STACK TO SE
-IF NB <<SF>> MOV (SP),SF ;POP STACK TO SF
-ENDM SPOP

```

```

313
314
315
316 000240* 016767 177550 177770 START: ;BEGIN TESTING FOR THE DUP-11
317 000246* 001002 MOV DVID1,SELECT ;GET ACTIVE DEVICES
318 000250* 104410 000000* DROP: BNE RESTR1 ;BR IF ANY ARE SELECTED
319 ENDS,BEGIN ;INCORRECT NUMBER OF DUP11'S
320
321 ;SETUP VECTORS FOR ACTIVE DEVICES
322 000254* 032767 177400 177754 RESTR1: BIT #C377,SELECT ;MAKE SURE NO MORE THAN 8 SELECTED.
323 000257* 001372 MOV DROP ;INVALID-NO MORE THAN 8 DEVICES!!
324 000264* 016701 177746 BNE SELECT,R1 ;GET IMAGE OF RUNNING DEVICES
325 000270* 001767 BEQ DROP ;BR IF ALL DUP'S DROPPED
326 000272* 005067 177726 CLR N,DUPS ;CLEAR THE # OF DUP'S (SOFTWARE)
327 000276* 016702 177506 MOV VECTOR,R2 ;GET INITIAL VECTOR
328 000302* 016700 MOV ADDR,R0 ;GET INITIAL ADDRESS
329 000306* 012703 001614* MOV #NKF0,R3 ;SET ISR POINTER
330 000312* 012767 002114* 177712 MOV #RXP0,R3F ;SET UP BUFFER POINTERS
331 000320* 006201 1$: ASR R1 ;ACTIVE?
332 000322* 006201 BCS 3$ ;BR IF YES
333 000324* 001437 BEQ SETUP1 ;BR IF DONE
334 000326* 062703 000024 ADD #24,R3 ;PCP ISR POINTER
335 000328* 000010 2$: ADD #10,R2 ;POP VECTOR
336 000336* 062700 000010 BR 1$ ;POP CSR
337 000342* 000766 BR 1$ ;CONTINUE
338 000344* 005267 177654 3$: INC N,DUPS ;CONTINUE THE # TO RUN
339 000346* 010312 MOV R3,(R2) ;LOAD ISR POINTER (RECEIVER)
340 000352* 116762 177434 000002 MOV B1,2(R2) ;LOAD PRIORITY
341 000360* 010063 000004 MOV R0,4(R3) ;LOAD CSR POINTER
342 000364* 005063 000010 CLR 10(R3) ;LOAD TX BYTE COUNT
343 000370* 006203 000010 ADD #12,R3 ;UPDATE POINTER
344 000374* 010362 000004 MOV R3,4(R2) ;LOAD TRANSMITTER ISR
345 000400* 116762 177407 000006 MOV B1,2(R2) ;LOAD PRIORITY
346 000406* 010063 000004 MOV R0,4(R3) ;LOAD CSR POINTER
347 000410* 006203 000010 CLR 10(R3) ;CLEAR TX BYTE COUNT
348 000416* 006203 000012 ADD #12,R3 ;POP ISR POINTER
349 000422* 000743 BR 2$ ;CONTINUE
350
351 ;SET UP BUFFERS
352 000424* 016767 177574 177466 SETUP1: MOV N,DUPS,INTR ;SET # OF INTERRUPTS
353 000432* 016767 177566 MOV DUPS,MDTO ;SET # OF WORDS TO MEM
354 000440* 016767 177560 177450 MOV N,DUPS,MDFR ;SET # OF WORDS FROM MEM
355 000446* 006367 ASL INTR ;DOUBLE INTERRUPTS
356 000450* 006367 MOV #8,R0 ;LOAD BUFFER SIZE
357 000455* 012703 002074* 1$: MOV #RXP0,R3 ;LOAD BUFFER START
358 000462* 012723 000135* BR 1$ ;UPDATE
359 000470* 001374 DEC R0 ;FOR FINISH
360 000472* 012700 000010 BNE 1$ ;BR IF MORE TO GO
361 000476* 012703 002114* MOV #10,R0 ;LOAD BUFFER SIZE
362 000480* 012703 000136 2$: MOV #RXP0,R3 ;LOAD START ADDRESS
363 000506* 005300 1$: MOV #136,(R3)+ ;CLEAR
364 000510* 001374 DEC R0 ;LOWER THE # TO DO
365 000512* 016700 177270 2$: BNE 2$ ;BR IF MORE TO GO
366
367 ;PRELIMINARY DEVICE SETUP
368 000512* 016700 177270 SETUP2: MOV ADDR,R0 ;LOAD FIRST CSR
  
```

```

369 000516* 016701 177514 1$: MOV SELECT,R1 ;SET UP TO GET ACTIVE DEVICES
370 000522* 006201 1$: ASR R1 ;SET UP TO GET ACTIVE DEVICES
371 000524* 103404 BCS 2$ ;BR IF DONE
372 000526* 001410 BEQ ACTIV ;BR IF DONE
373 000530* 062700 000010 ADD #10,R0 ;UPDATE FOR THE NEXT ONE
374 000534* 000772 BR 1$ ;UPDATE FOR THE NEXT ONE
375 000536* 004767 000766 2$: JSR PC,PRELIM ;GO DO THE DEVICE SETUP
376 000542* 062700 000010 BR 1$ ;UPDATE FOR THE NEXT ONE
377 000546* 000765 BR 1$ ;CONTINUE
378 000550* 016701 177452 ACTIV: MOV SELECT,R1 ;GET THE ACTIVE DUP'S
379 000554* 016767 177444 MOV N,DUPS,TOTAL ;SET UP FOR DATA CHECK
380 000562* 016700 177220 MOV ADDR,R0 ;GET FIRST CSR
381 000566* 006201 1$: ASR R1 ;GET AN ACTIVE ONE
382 000570* 103404 BCS 3$ ;BR IF ACTIVE
383 000572* 001406 BEQ WAIT ;BR IF DONE
384 000574* 062700 000010 ADD #10,R0 ;UPDATE CSR
385 000600* 000772 000764 3$: JSR PC,READY ;TURN ON DEVICE
386 000606* 000772 BR 2$ ;CONTINUE
387
388 ;DELAY AND SCAN FOR FINISH ROUTINE
389 000610* 005005 WAIT: CLR R5 ;SET FOR A LONG DELAY
390 000612* 012704 000002 MOV #2,R4 ;APPROX 30 SECONDS
391 000616* 000000 1$: SCAN: MOV #2,R4 ;APPROX 30 SECONDS
392 000616* 000000
393 000616* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
394 000622* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
395 000626* 005767 177374 BNE 3$ ;GET THE # OF ACTIVE DUPS
396 000632* 001004 BR 1$ ;BR IF MORE TO GO
397
398 -----
399 000634* 104413 000000* ENDT1$,BEGIN ;SIGNAL END OF ITERATION.
400 ;MONITOR SHALL TEST END OF PASS
401 000640* 000167 177410 2$: JMP RESTR1 ;RESTART PROGRAM
402 000644* 005305 3$: DEC R5 ;STALL FOR ALL DUP'S TO FINISH
403 000646* 001363 BNE 1$ ;BR IF DELAY NOT ZERO
404 000650* 105304 DECB R4 ;DO ANOTHER DELAY
405 000654* 001361 BNE 1$ ;BR IF MORE TO GO
406 000658* 104403 000000* 000666* MSGN$,BEGIN,HUNG ;ASCII MESSAGE CALL WITH COMMON HEADER
407 000662* 104410 000000* ENDS$,BEGIN ;DROP THE MODULE
408 000666* 002154* HUNG: MOV #1,MSGN$ ;POINT TO ASCII MESSAGE
409 000670* 177777 -1 ;TERMINATOR
  
```

```

411
412 000672* 010046 TXISR: ;TRANSMITTER INTERRUPT SERVICE ROUTINE
413 000672* 010146 MOV RO,-(SP) ;SAVE RO ON THE STACK
414 000674* 010146 MOV RI,(SP) ;SAVE RI ON THE STACK
415 000676* 012500 MOV (R5)+,RO ;GET DEVICE CSR AND POP OFFSET TO R5
416 000700* 012501 MOV (R5)+,R1 ;GET DATA ADDRESS AND LEAVE R5=COUNT ADDRESS
417 000700* 032760 100000 000004 BIT TXDLAT,TXCSR(RO) ;CHECK FOR OVERFLOW?
418 000710* 001427 BEQ 5S ;BR IF NO
419 000711* 010067 177162 MOV RO,CSRA ;LOAD FOR PRINTOUT
420 000716* 005067 177160 CLR ACSR ;CLEAR
421 000722* 012760 000400 000004 MOV #MRESET,TXCSR(RO) ;TURN OFF DEVICE
422 000730* 005367 177272 DEC TOTAL ;DECREMENT THE # TO DO
423 000734* 012601 MOV (SP)+,R1 ;POP STACK TO R1
424 000736* 012600 MOV (SP)+,RO ;POP STACK TO RO
425 000740* 012605 MOV (SP)+,R5 ;POP STACK TO R5
426
427 000742* 000004 000000* 000750* ;PIRQS,BEGIN,1S ;QUEUE UP TO CONTINUE AT 1S AND RTI
428
429 000750* 012767 000041 177130 1S: MOV #41,ERRTYP ;XMITTER DATA LATE
430
431 000756* 104406 000000* 000000 ;***** ;*****
432 ;***** ;*****
433 ;***** ;*****
434 ;***** ;*****
435 ;***** ;*****
436 ;***** ;*****
437 ;***** ;*****
438 ;***** ;*****
439 ;***** ;*****
440 ;***** ;*****
441 ;***** ;*****
442 ;***** ;*****
443 ;***** ;*****
444 ;***** ;*****
445 ;***** ;*****

```

```

446
447 001032* 010046 RXISR: ;RECEIVER INTERRUPT SERVICE ROUTINE
448 001032* 010146 MOV RO,-(SP) ;SAVE RO ON THE STACK
449 001034* 010146 MOV RI,(SP) ;SAVE RI ON THE STACK
450 001036* 012500 MOV (R5)+,RO ;GET DEVICE CSR AND POP OFFSET TO R5
451 001040* 012501 MOV (R5)+,R1 ;GET DATA ADDRESS AND LEAVE R5=BYTE COUNT
452 001042* 105710 TSTB (RO) ;CHECK FOR DONE
453 001044* 100427 BMI 2S ;BR IF SET
454 001046* 010067 177026 MOV RO,CSRA ;SETUP FOR TYPEOUT
455 001052* 011067 177024 MOV #MRESET,TXCSR(RO) ;TURN OFF DEVICE
456 001056* 012760 000400 000004 DEC TOTAL ;DECREMENT THE # TO DO
457 001064* 005367 177136 MOV (SP)+,R1 ;POP STACK TO R1
458 001070* 012601 MOV (SP)+,RO ;POP STACK TO RO
459 001072* 012600 MOV (SP)+,R5 ;POP STACK TO R5
460 001074* 012605
461
462 001076* 000004 000000* 001104* ;PIRQS,BEGIN,1S ;QUEUE UP TO CONTINUE AT 1S AND RTI
463
464 001104* 012767 000011 176774 1S: MOV #11,ERRTYP ;ILLEGAL INTERRUPT
465
466 001112* 104405 000000* 000000 ;***** ;*****
467 ;***** ;*****
468 ;***** ;*****
469 ;***** ;*****
470 ;***** ;*****
471 ;***** ;*****
472 ;***** ;*****
473 ;***** ;*****
474 ;***** ;*****
475 ;***** ;*****
476 ;***** ;*****
477 ;***** ;*****
478 ;***** ;*****
479 ;***** ;*****
480 ;***** ;*****
481 ;***** ;*****
482 ;***** ;*****
483 ;***** ;*****
484 ;***** ;*****
485 ;***** ;*****
486 ;***** ;*****
487 ;***** ;*****
488
489 001220* 000004 000000* 001226* ;PIRQS,BEGIN,4S ;QUEUE UP TO CONTINUE AT 4S AND RTI
490
491 001226* 012767 000017 176652 4S: MOV #17,ERRTYP ;OVERRUN OR CRC ERROR
492
493 001234* 104405 000000* 000000 ;***** ;*****
494 ;***** ;*****
495 ;***** ;*****
496 ;***** ;*****
497 ;***** ;*****
498 ;***** ;*****
499 ;***** ;*****
500 ;***** ;*****
501 ;***** ;*****

```

```

502 001274* 012605          MOV (SP)+,R5 ;POP STACK TO R5
503
504 001276* 000004 000000* 001304*  PIRQS,BEGIN,6S ; QUEUE UP TO CONTINUE AT 6S AND RTI
505
506 001304* 012767 000042 176574 6S: MOV #42,ERRTYP ;ACTIVE CLEARED AFTER DONE
507 *****;*****
508 001312* 104405 000000* 000000  HDRERS,BEGIN,NULL ;ACTIVE CLEARED AFTER A DONE
509 *****;*****
510 001320* 000167 177272          JMP SCAN ;CONTINUE SCANNING FOR END
511 001324* 010167 176552          MOV R1,SBADR ;LOAD GOOD DATA ADRS
512 001330* 010067 176544          MOV R0,CSRA ;LOAD CSR
513 001332* 012767 000234* 176542  MOV #SAVBF,WASADR ;LOAD BAD DATA ADRS
514 001334* 111167 176540          MOV (R1),ASB ;LOAD GOOD DATA
515 001346* 116767 176662 176534  NOVB SAVBF,WAS ;LOAD BAD DATA
516 001354* 012760 000400 000004  MOV #MRESET,TXCSR(R0) ;TURN OFF DEVICE
517 001362* 005367 176640          DEC TOTAL ;LOWER THE # TO DO
518 001370* 012600          MOV (SP)+,R1 ;POP STACK TO R1
519 001372* 012605          MOV (SP)+,R0 ;POP STACK TO R0
520 001374* 000004 000000* 001402*  MOV (SP)+,R5 ;POP STACK TO R5
521 -----
522 001402* 000004 000000* 001402*  PIRQS,BEGIN,10S ; QUEUE UP TO CONTINUE AT 10S AND RTI
523 -----
524 001402* 000004 000000* 10S: ;*****
525 *****;*****
526 *****;*****
527 *****;*****
528 *****;*****
529 *****;*****
530 *****;*****
531 *****;*****
532 *****;*****
533 *****;*****
534 *****;*****
535 *****;*****
536 *****;*****
537 *****;*****
538 *****;*****
539 *****;*****
540 *****;*****
541 *****;*****
542 *****;*****
543 *****;*****
544 *****;*****
545 *****;*****
546 *****;*****
547 *****;*****
548 *****;*****
549 *****;*****
550 *****;*****
551 *****;*****
552 *****;*****
553 *****;*****
554 *****;*****
555 *****;*****
556 *****;*****
557 *****;*****
558 *****;*****
559 *****;*****
560 *****;*****
561 *****;*****
562 *****;*****
563 *****;*****
564 *****;*****
565 *****;*****
566 *****;*****
567 *****;*****
568 *****;*****
569 *****;*****
570 *****;*****
571 *****;*****
572 *****;*****
573 *****;*****
574 *****;*****
575 *****;*****
576 *****;*****
577 *****;*****
578 *****;*****

```

```

548 001506* 012760 000400 000004 13S: MOV #MRESET,TXCSR(R0) ;TURN OFF DEVICE
549 001514* 005367 176506          DEC TOTAL ;LOWER THE # TO DO
550
551 001520* 012601          MOV (SP)+,R1 ;POP STACK TO R1
552 001522* 012600          MOV (SP)+,R0 ;POP STACK TO R0
553 001524* 012605          MOV (SP)+,R5 ;POP STACK TO R5
554 001526* 000002          RTI ;RETURN
555
556
557
558
559
560
561
562
563
564
565 001530* 052760 000400 000004  PRELIM: ;ROUTINE TO TURN ON DUP'S
566 001536* 004767 000046          BIS #MRESET,TXCSR(R0) ;RESET THE DUP11
567 001542* 052760 004000 000004  JSR PC,SMALL ;WAIT FOR RESET TO END
568 001550* 052760 010135 000002  BIS #SYSTST,TXCSR(R0) ;TURN ON SYSTEM TEST MODE
569 001556* 052710 000120          BIS #PRISEC135,PARCSR(R0) ;ENTER SECONDARY MODE AND ADDRESS
570 001562* 052760 000020 000004  BIS #RCVEMIRINTEN(R0) ;TURN ON RECEIVER AND INT. ENABLE
571 001570* 000406          BR RETURN ;TURN ON TRANSMITTER
572 001572* 052760 000400 000006  READY: BIS #TSON,TXDBUF(R0) ;JUMP AROUND
573 001600* 052760 000100 000004  RETURN: BIS #XKINTE,TXCSR(R0) ;TURN ON TRANSMITTER INT. ENABLE
574 001606* 000207          RTS PC ;RETURN
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678

```



```

579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634

```

001614\*

```

LNKTAB:
.MACRO
    $JS,$Q
    JSR      R5,RXISR
    0
    RXBF*$Q
    BCNT*$Q
    JSR      R5,TXISR
    0
    TXBF*$Q
    OFSET*$Q
.ENDM
    $JS
    JSR      R5,RXISR
    0
    RXBF0
    BCNT0
    JSR      R5,TXISR
    0
    TXBF0
    OFSET0
    JSR      R5,RXISR
    0
    RXBF1
    BCNT1
    JSR      R5,TXISR
    0
    TXBF1
    OFSET1
    JSR      R5,RXISR
    0
    RXBF2
    BCNT2
    JSR      R5,TXISR
    0
    TXBF2
    OFSET2
    JSR      R5,RXISR
    0
    RXBF3
    BCNT3
    JSR      R5,TXISR
    0
    TXBF3
    OFSET3

```

ANSWER FOR DEVICE \*SQ REC ISR  
CSR FOR DEVICE \*SQ REC ISR  
BUFFER LOC FOR DEVICE \*SQ  
BYTE COUNT ADDRESS FOR RXBF\*SQ  
ANSWER FOR DEVICE \*SQ TX ISR  
CSR FOR DEVICE \*SQ TX ISR  
DATA ADDRESS FOR DEVICE \*SQ  
BYTE COUNT ADDRESS FOR DEVICE \*SQ  
ANSWER FOR DEVICE 0 REC ISR  
CSR FOR DEVICE 0 REC ISR  
BUFFER LOC FOR DEVICE 0  
BYTE COUNT ADDRESS FOR RXBF0  
ANSWER FOR DEVICE 0 TX ISR  
CSR FOR DEVICE 0 TX ISR  
DATA ADDRESS FOR DEVICE 0  
BYTE COUNT ADDRESS FOR DEVICE 0  
ANSWER FOR DEVICE 1 REC ISR  
CSR FOR DEVICE 1 REC ISR  
BUFFER LOC FOR DEVICE 1  
BYTE COUNT ADDRESS FOR RXBF1  
ANSWER FOR DEVICE 1 TX ISR  
CSR FOR DEVICE 1 TX ISR  
DATA ADDRESS FOR DEVICE 1  
BYTE COUNT ADDRESS FOR DEVICE 1  
ANSWER FOR DEVICE 2 REC ISR  
CSR FOR DEVICE 2 REC ISR  
BUFFER LOC FOR DEVICE 2  
BYTE COUNT ADDRESS FOR RXBF2  
ANSWER FOR DEVICE 2 TX ISR  
CSR FOR DEVICE 2 TX ISR  
DATA ADDRESS FOR DEVICE 2  
BYTE COUNT ADDRESS FOR DEVICE 2  
ANSWER FOR DEVICE 3 REC ISR  
CSR FOR DEVICE 3 REC ISR  
BUFFER LOC FOR DEVICE 3  
BYTE COUNT ADDRESS FOR RXBF3  
ANSWER FOR DEVICE 3 TX ISR  
CSR FOR DEVICE 3 TX ISR  
DATA ADDRESS FOR DEVICE 3  
BYTE COUNT ADDRESS FOR DEVICE 3

```

635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675

```

```

    JSR      R5,RXISR
    0
    RXBF4
    BCNT4
    JSR      R5,TXISR
    0
    TXBF4
    OFSET4
    JSR      R5,RXISR
    0
    RXBF5
    BCNT5
    JSR      R5,TXISR
    0
    TXBF5
    OFSET5
    JSR      R5,RXISR
    0
    RXBF6
    BCNT6
    JSR      R5,TXISR
    0
    TXBF6
    OFSET6
    JSR      R5,RXISR
    0
    RXBF7
    BCNT7
    JSR      R5,TXISR
    0
    TXBF7
    OFSET7

```

ANSWER FOR DEVICE 4 REC ISR  
CSR FOR DEVICE 4 REC ISR  
BUFFER LOC FOR DEVICE 4  
BYTE COUNT ADDRESS FOR RXBF4  
ANSWER FOR DEVICE 4 TX ISR  
CSR FOR DEVICE 4 TX ISR  
DATA ADDRESS FOR DEVICE 4  
BYTE COUNT ADDRESS FOR DEVICE 4  
ANSWER FOR DEVICE 5 REC ISR  
CSR FOR DEVICE 5 REC ISR  
BUFFER LOC FOR DEVICE 5  
BYTE COUNT ADDRESS FOR RXBF5  
ANSWER FOR DEVICE 5 TX ISR  
CSR FOR DEVICE 5 TX ISR  
DATA ADDRESS FOR DEVICE 5  
BYTE COUNT ADDRESS FOR DEVICE 5  
ANSWER FOR DEVICE 6 REC ISR  
CSR FOR DEVICE 6 REC ISR  
BUFFER LOC FOR DEVICE 6  
BYTE COUNT ADDRESS FOR RXBF6  
ANSWER FOR DEVICE 6 TX ISR  
CSR FOR DEVICE 6 TX ISR  
DATA ADDRESS FOR DEVICE 6  
BYTE COUNT ADDRESS FOR DEVICE 6  
ANSWER FOR DEVICE 7 REC ISR  
CSR FOR DEVICE 7 REC ISR  
BUFFER LOC FOR DEVICE 7  
BYTE COUNT ADDRESS FOR RXBF7  
ANSWER FOR DEVICE 7 TX ISR  
CSR FOR DEVICE 7 TX ISR  
DATA ADDRESS FOR DEVICE 7  
BYTE COUNT ADDRESS FOR DEVICE 7

```
676 ;BUFFER AREAS
677
678
679 .MACRO $OF,$Q
680 OFSET0: .WORD 0 ;TRANSMITTER BYTE COUNT POINTER FOR DEVICE *$Q
681 .ENDM
682 OFSET1: .WORD 0 ;TRANSMITTER BYTE COUNT POINTER FOR DEVICE 0
683 OFSET2: .WORD 0 ;TRANSMITTER BYTE COUNT POINTER FOR DEVICE 1
684 OFSET3: .WORD 0 ;TRANSMITTER BYTE COUNT POINTER FOR DEVICE 2
685 OFSET4: .WORD 0 ;TRANSMITTER BYTE COUNT POINTER FOR DEVICE 3
686 OFSET5: .WORD 0 ;TRANSMITTER BYTE COUNT POINTER FOR DEVICE 4
687 OFSET6: .WORD 0 ;TRANSMITTER BYTE COUNT POINTER FOR DEVICE 5
688 OFSET7: .WORD 0 ;TRANSMITTER BYTE COUNT POINTER FOR DEVICE 6
689
690
691 .MACRO $TB,$Q
692 TXBF*$Q: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE *$Q
693 .ENDM
694 TXBF0: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE 0
695 TXBF1: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE 1
696 TXBF2: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE 2
697 TXBF3: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE 3
698 TXBF4: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE 4
699 TXBF5: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE 5
700 TXBF6: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE 6
701 TXBF7: .WORD 0 ;TRANSMITTER BUFFER FOR DEVICE 7
702
703
704 .MACRO $RB,$Q
705 RXBF*$Q: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #*$Q
706 .ENDM
707 RXBF0: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #0
708 RXBF1: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #1
709 RXBF2: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #2
710 RXBF3: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #3
711 RXBF4: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #4
712 RXBF5: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #5
713 RXBF6: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #6
714 RXBF7: .WORD 0 ;RECEIVER BUFFER FOR DEVICE #7
715
716
717 .MACRO $BC,$Q
718 BCNT*$Q: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE *$Q
719 .ENDM
720 BCNT0: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE 0
721 BCNT1: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE 1
722 BCNT2: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE 2
723 BCNT3: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE 3
724 BCNT4: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE 4
725 BCNT5: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE 5
726 BCNT6: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE 6
727 BCNT7: .WORD 0 ;RECEIVER BYTE COUNT POINTER FOR DEVICE 7
728
729
730
731
```

;ASCII MESSAGES

```
732 002154* 042045 050125 030461 MHUNG: .ASCIZ "$DUP11 MODULE IS HUNG - SEE LISTING?"
733 002227 .EVEN
000001 .END
```





DPBB DEC/X11 SYSTEM EXERCISER MODULE  
XDPBBO.P11 12-OCT-78 11:53

MACY11 30A(1052) 12-OCT-78 16:30 PAGE 24  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0022

XDPBBO, XDPBBO/SOL/CRF:SYM=DDXCOM, XDPBBO  
RUN-TIME: 22.4 SECONDS  
RUN-TIME RATIO: 17/5=2.5  
CORE USED: 7K (13 PAGES)