

.REM ^

IDENTIFICATION

PRODUCT CODE: AC-E947B-MC
PRODUCT NAME: CXDVAB0 DV11 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 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) 1976, 1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

DVA IS AN IOMOD THAT EXERCISES UP TO AND INCLUDING FOUR (CONSECUTIVELY ADDRESSED) DV11 SYNCHRONOUS INTERFACES. IT USES MAINTAINANCE MODE TO XMIT AND RECEIVE A SET DATA PATTERN RESEMBLING A BINARY COUNT FROM 00-77. THE RECEIVER AND TRANSMITTER ISR ARE ALWAYS AT PRIORITY FIVE. (BR1, BR2) DATA CHECKING IS PERFORMED AT LEVEL 0 AND DONE OUTSIDE THE ISRS.

2. REQUIREMENTS

HARDWARE: AT LEAST 1 DV11-AA AND ONE DV11-BA

STORAGE: DVA REQUIRES:

- 1. DECIMAL WORDS: 3939
- 2. OCTAL WORDS: 07543
- 3. OCTAL BYTES: 17306

3. PASS DEFINITION

ONE PASS OF THE DVA MODULE CONSISTS OF HANDLING 100(8) CHARS TIMES THE NUMBER OF SELECTED LINES TIMES THE NUMBER OF SELECTED DEVICES TIMES 600(8).

4. EXECUTION TIME

THIS VARIES DUE TO THE CONFIGURATION BUT SHOULD ALWAYS BE UNDER 01 MIN.

5. CONFIGURATION PARAMETERS.

DEFAULT PARAMETERS:
ADDR: 175000, VECTOR: 310, BR1: 5, BR2: 5, DVID1: 1, SR1: 0
FIRST EIGHT LINES FOR EACH DV11.
SYNC "A" SET TO 226
USER *MUST* SET SOFTWARE "SYNC" TO "377"
FOR TEST OF AN ASYNC LINE CARD. SEE SECTION 8.
USER MAY ALTER PARAMETERS. NO MORE THAN 4 DV11'S MAY BE RUN.

6. DEVICE OPTION SETUP

- 1. PARITY OFF
 - 2. AT LEAST 6 BITS PER CHAR
 - 3. SYNC "A" OVER(GREATER THAN) 20(8)
 - 4. FULL DUPLEX
- DEFAULT SHIP OF 8 BITS, PARITY OFF, FULL DUPLEX, ETC OK

7. MODULE OPERATION

1. LOAD SOFTWARE POINTERS IN LNKTABLE.
2. LOAD VECTORS AND PRIORITIES IN TABLE.
3. DETERMINE WHICH SET OF LINES TO TEST (PING.PONG = 0 LINES 00-07; PING.MASTER CLEAR DEVICE.
4. CLEAR ALL SECONDARY REGISTERS FOR SELECTED LINES.
5. LOAD ALL SECONDARY REGISTERS FOR SELECTED LINES.
6. UPDATE ALL LINE COUNTER IF NOT DONE GOTO 5.
7. LOAD INDIVIDUAL SECONDARY REGISTERS THAT VARY FROM LINE TO LINE OR DEVICES.
8. ENABLE FOR ALL LINES OF ALL DEVICES TO FINISH.
9. SCAN FOR ALL LINES OF ALL DEVICES TO FINISH.
10. IF NOT DONE GOTO 9.
11. HUNG REPORT SO AND DROP MODULE FOR ALL DEVICES SELECTED.
12. CHECK DATA FOR ALL LINES SELECTED FOR ALL DEVICES SELECTED.
13. SWITCH TO NEXT GROUP OF LINES (COM PING.PONG)
14. DECREMENT ITERATION COUNT
15. IF NOT = 0 GOTO 1
16. SIGNAL ENDPAS. TRANSMITTER INTERRUPT SERVICE ROUTINE.
17. TXISR. TRANSMITTER INTERRUPT SERVICE ROUTINE.
18. GET INTERRUPTING DVSCR.
19. WAS DVSCR 15=1? (IF NOT; REPORT ERROR)
20. READ DVNSR
21. CHECK FOR EITHER PRI BC OR ALT BC = 0
22. RTI
23. RXISR; RECEIVER INTERRUPT SERVICE ROUTINE.
24. GET INTERRUPTING DVSCR
25. WAS DVSCR 7=1? (IF NOT; REPORT ERROR)
26. ARE ANY ERROR CODES SET? THIS SPECIAL CHAR 77?
27. IF YES (CHAR = 77) THEN SHUT TX OFF AND RESYNC RX. (SIGNAL END OF LINE ACTIVITY)
28. IF ERROR CODE PRESENT THEN WAS IT BCC CODE?
29. IF NO THEN REPORT ERROR. IF YES WAS BCC = 0? (IF NOT; REPORT ERROR)
30. RTI

8. OPERATOR OPTIONS

LOCATIONS IN DVA FROM 164(8) TO 232(8) ARE BASICALLY SELF EXPLANATORY AND MAY ALTERED SPECIFIC DVIL CONFIGURATION. VIA THE MOD CMD TO ALTER TO THE SPECIFIC DVIL CONFIGURATION. NOTE: IF YOU HAVE MORE THAN 8 LINES INSTALLED, IT IS YOUR RESPONSIBILITY TO ALTER THE "LINES(1,2,3,4) LOCATIONS TO HAVE THESE LINES TESTED. ASYNCHRONOUS LINE CARD TESTING. FOR TESTING OF ASYNC LINE CARDS; BOTH SYNC CHARS FOR THAT LINE CARD *MUSI* BE SET TO "377" (WORD=17777) IN THIS PROGRAM. LOOK AT LOCATIONS 164(8) THRU 232(8) FOR CORRECT LOCATION OF SYNC CHARS FOR LINE CARD.

9. NON-STANDARD PRINTOUTS

IF THE MODULE "HANGS" IN WHICH NOT ALL LINES FOR EACH DVIL SELECTED FAIL TO FINISH, A "HUNG" MESSAGE IS PRINTED OUT. TO FIND OUT WHAT LINE(S) ON WHICH DEVICE(S) HAVE FAILED; COMPARE THE LINES SELECTED BETWEEN 164(8)-222(8) TO THE IMAGE BEING CREATED AT THE "LNKTAB" ARGUMENT FOLLOWING THE DYSCR IMAGE OF THE TRANSMITTER "JSR" ROUTINE.

THAT IS TO SAY:

LINES1(164) SHOULD EQUAL (COMPARE TO) LOC: XXX1 016742
LINES2(176) SHOULD EQUAL (COMPARE TO) LOC: XXX2 016762
LINES3(210) SHOULD EQUAL (COMPARE TO) LOC: XXX3 017002
LINES4(222) SHOULD EQUAL (COMPARE TO) LOC: XXX4 017022

FOR A COMPLETE RUN OF DVA. IF ANY OF THE FOLLOWING LOCATIONS FAIL TO COMPARE A "HUNG" MESSAGE WILL BE PRINTED AND THE MODULE WILL BE DROPPED.

```
217 000000* IOMOD <DVAB > 175000,310,5,5,0,600,74
218 000000* MODULE 140000,DVAB,175000,BID,5,5,0,600,74
219 ; TITLE DVAB DEC/X11 SYSTEM EXERCISER MODULE
220 ; DDXCOM VERSION 6 23-MAY-78
221 ; LIST RH
222 ;*****
223 BEGIN:
224 MODNAM: .ASCII /DVAB / ;MODULE NAME
225 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WRUFF USAGE
226 ADDR: 175000+0 ;1ST DEVICE ADDR
227 VECTOR: 310+0 ;1ST DEVICE VECTOR.
228 BR1: .BYTE PRTV5+0 ;1ST BR LEVEL.
229 BR2: .BYTE PRTV5+0 ;2ND BR LEVEL.
230 DVID1: 0+1 ;DEVICE INDICATOR 1.
231 SR1: OPEN ;SWITCH REGISTER 1.
232 SR2: OPEN ;SWITCH REGISTER 2
233 SR3: OPEN ;SWITCH REGISTER 3
234 SR4: OPEN ;SWITCH REGISTER 4
235 ;*****
236 STAT: 140000 ;STATUS WORD.
237 INIT: START ;MODULE START ADDR.
238 SPOINT: MODSP ;MODULE STACK POINTER.
239 PASCNT: 0 ;PASS COUNTER.
240 ICONT: 600 ;# OF ITERATIONS PER PASS=600
241 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
242 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
243 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
244 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
245 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
246 SYSENR: 0 ;# OF SYS ERRORS ACCUMULATED
247 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRC IS CALLED
248 CONFIG:
249 RES1: 0 ;RESERVED FOR MONITOR USE
250 RES2: 0 ;RESERVED FOR MONITOR USE
251 SVRO: OPEN ;LCC TO SAVE R0.
252 SVR1: OPEN ;LCC TO SAVE R1.
253 SVR2: OPEN ;LCC TO SAVE R2.
254 SVR3: OPEN ;LCC TO SAVE R3.
255 SVR4: OPEN ;LCC TO SAVE R4.
256 SVR5: OPEN ;LCC TO SAVE R5.
257 SVR6: OPEN ;LCC TO SAVE R6.
258 SVR7: OPEN ;LCC TO SAVE R7.
259 SBADR: OPEN ;ADDR OF CURRENT CSR.
260 ACSR: OPEN ;ADDR OF GOOD DATA, OR
261 WASADR: OPEN ;ADDR OF BAD DATA, OR
262 ASR: OPEN ;STATUS REG CONTENTS.
263 ERRTYP: OPEN ;TYPE OF ERROR
264 ASB: OPEN ;EXPECTED DATA.
265 AWAS: OPEN ;ACTUAL DATA.
266 WADR: RESRST ;RESTART ADDRESS AFTER END OF PASS
267 WDRS: OPEN ;WCRDS TO MEMORY PER ITERATION
268 WDFR: OPEN ;WCRDS FROM MEMORY PER ITERATION
269 INTI: OPEN ;# OF INTERRUPTS PER ITERATION
270 INUM: 74 ;MODULE IDENTIFICATION NUMBER=74
271 MODSP:
272 ;*****
```

```
273 ;*SETUP VARIABLES NEEDED FOR DV11 CONFIGURATION
274
275 000224* 000377 LINES1: ^B<0000000011111111> ;DEFAULT ALL 8 LINES
276 000226* 226 226 SYNC11: .BYTE 226,226 ;DEFAULT SYNC A (1)
277 000230* 226 226 SYNC12: .BYTE 226,226 ;DEFAULT SYNC A (2)
278 000234* 226 226 SYNC13: .BYTE 226,226 ;DEFAULT SYNC A (3)
279 000234* 226 226 SYNC14: .BYTE 226,226 ;DEFAULT SYNC A (4)
280
281 000236* 000377 LINES2: ^B<0000000011111111> ;DEFAULT ALL 8 LINES
282 000240* 226 226 SYNC21: .BYTE 226,226 ;DEFAULT SYNC A (1)
283 000242* 226 226 SYNC22: .BYTE 226,226 ;DEFAULT SYNC A (2)
284 000244* 226 226 SYNC23: .BYTE 226,226 ;DEFAULT SYNC A (3)
285 000246* 226 226 SYNC24: .BYTE 226,226 ;DEFAULT SYNC A (4)
286
287 000250* 000377 LINES3: ^B<0000000011111111> ;DEFAULT ALL 8 LINES
288 000252* 226 226 SYNC31: .BYTE 226,226 ;DEFAULT SYNC A (1)
289 000254* 226 226 SYNC32: .BYTE 226,226 ;DEFAULT SYNC A (2)
290 000254* 226 226 SYNC33: .BYTE 226,226 ;DEFAULT SYNC A (3)
291 000260* 226 226 SYNC34: .BYTE 226,226 ;DEFAULT SYNC A (4)
292
293 000262* 000377 LINES4: ^B<0000000011111111> ;DEFAULT ALL 8 LINES
294 000264* 226 226 SYNC41: .BYTE 226,226 ;DEFAULT SYNC A (1)
295 000266* 226 226 SYNC42: .BYTE 226,226 ;DEFAULT SYNC A (2)
296 000270* 226 226 SYNC43: .BYTE 226,226 ;DEFAULT SYNC A (3)
297 000272* 226 226 SYNC44: .BYTE 226,226 ;DEFAULT SYNC A (4)
298
299 000000 DVSCR=0 ;SYSTEM CONTROL REGISTER
300 000002 DVICR=2 ;RECEIVER INTERRUPT CHAR REGISTER
301 000004 DVLCR=4 ;LINE CONTROL REGISTER
302 000006 DVSR5=6 ;SECONDARY REGISTER SELECT REGISTER
303 000007 DVSRSH=7 ;SECONDARY REGISTER SELECT REGISTER (HIGH BYTE)
304 000010 DVSRN=10 ;SECONDARY REGISTER ACCESS REGISTER
305 000012 DVSRF=12 ;SPECIAL FUNCTIONS REGISTER
306 000014 DVNSR=14 ;NPR STATUS REGISTER
307 000016 DVRES=16 ;RESERVED REGISTER
308
309 000000 MODE0=0*40
310 000040 MODE1=1*40
311 000100 MODE2=2*40
312 000140 MODE3=3*40
313 000200 MODE4=4*40
314 000240 MODE5=5*40
315 000300 MODE6=6*40
316 000340 MODE7=7*40
317
318 000274* 000001 PING.PONG: .BLKW 1
319 000276* 000001 VA: .BLKW 1
320 000300 000001 PA: .BLKW 1
321 000302 000001 EA: .BLKW 1
322 000304 000001 SELECT: .BLKW 1
323 000306 000001 XCNT: .BLKW 1
324 000310 000001 HOLD: .BLKW 1
325 000312 000001 STORE: .BLKW 1
```

```
325 ; BEGIN THE TESTS FOR THE DV11
326 ;
327 ;
328 ;
329 ;
330 ;
331 ;
332 ;
333 ;
334 ;
335 ;
336 ;
337 ;
338 ;
339 ;
340 ;
341 ;
342 ;
343 ;
344 ;
345 ;
346 ;
347 ;
348 ;
349 ;
350 ;
351 ;
352 ;
353 ;
354 ;
355 ;
356 ;
357 ;
358 ;
359 ;
360 ;
361 ;
362 ;
363 ;
364 ;
365 ;
366 ;
367 ;
368 ;
369 ;
370 ;
371 ;
372 ;
373 ;
374 ;
375 ;
376 ;
377 ;
378 ;
379 ;
380 ;
381 ;

000314 012767 002000 177572 START: MOV #2000,WDT0 ;2000 WORDS TO MEM/ITERATION
000322 012767 000000 177589 MOV #000,WDR ;2000 WORDS FROM MEM/ITERATION
000330 012767 000000 177589 MOV #000,WDR ;2000 WORDS FROM MEM/ITERATION
000336 005067 177322 RESTR: CLR ABORT ;CLEAR ABORT SERVICE FLAG
000342 005067 001534 177442 MOV WDI1,SELECT ;SET ACTIVE DEVICES
000346 018767 177442 BNE ;BR IF INV *ARE* SELECTED
000356 001002 000000 DROP: ENDS,BEGIN ;NO DV11'S SELECTED!
000356 104410 000000 CONT: BIT #<<17>,SELECT ;MAKE SURE NO MORE THAN 4 DEVICES SELECTED,
000370 001372 177760 177714 BNE ;ABSOLUTLY INVALID, NO MORE THAN 4 DEVICES!!
000372 016701 177706 SETUP1: MOV SELECT,R1 ;GET IMAGE OF RUNNING DEVICES
000376 016702 177404 DROP ;BR IF ALL DV11'S DROPPED
000400 016702 177404 MOV VECTOR,R2 ;SET INITIAL VECTOR
000404 016700 177376 MOV ADDR,R0 ;SET INITIAL CSR
000410 012703 047006 MOV #LNKTAB,R3 ;SET POINTER FOR ISR
000414 016763 177894 MOV LINES,4.(R3) ;LOAD DEFAULT ACTIVE LINES
000430 016763 177810 MOV LINES,30.(R3) ;LOAD DEFAULT ACTIVE LINES
000436 016763 177614 MOV LINES,46.(R3) ;LOAD DEFAULT ACTIVE LINES
000444 016763 177620 MOV LINES,62.(R3) ;LOAD DEFAULT ACTIVE LINES
000444 006201 1$: ASR R5 ;ACTIVE?
000448 013410 1$: BCS ;BR IF YES
000450 001440 1$: BEQ ;BR IF DONE
000452 062702 000020 2$: ADD #20,R2 ;POP VECTOR POINTER (20)
000456 062702 000020 2$: ADD #40,R0 ;POP SOFTWARE ISR POINTER (20)
000466 000766 000040 2$: ADD #40,R0 ;POP CSR POINTER (40)
000470 010372 1$: CONTINUE ;CONTINUE
000474 010372 177314 000002 3$: MOV R3,(R2) ;LOAD ISR POINTER (RECV)
000478 010372 000004 3$: MOV R0,4.(R3) ;LOAD PRIORITY
000482 010372 000010 3$: MOV R3,4.(R2) ;LOAD DVSCR POINTER
000486 010372 000010 3$: ADD #R0,4.(R2) ;LOAD ISR POINTER (TRAN)
000490 010372 000014 3$: MOV R0,4.(R2) ;POP I
000494 010372 000014 3$: MOV R0,4.(R3) ;LOAD DVSCR POINTER
000498 000530 177540 TST PING.PONG ;WHICH SET OF LINES??
000504 001003 000016 4$: BNE ;BR SECOND SET OF LINES
000508 001003 000016 4$: CLR B ;ZERO LOW LINES SELECTED
000512 001003 000016 4$: BR ;CONT
000516 001003 000016 4$: CLR B ;ZERO HIGH LINES SELECTED
000520 000740 2$: CONTINUE ;CONTINUE
000524 015700 177230 000224 SETUP2: MOV ADDR,R0 ;INITIAL DVSCR
000528 000562 000224 MOV #LINES1,R1 ;SET LINES POINTER
000532 000562 177506 TST PING.PONG ;WHICH SET OF LINES?
000536 001401 1$: BNE ;BR IF LOW SET OF LINES SELECTED
000540 001401 177510 1$: INC ;MAKE EQUAL TO HIGH BYTE
000544 010503 000017 MOV #BUFFER.TABLE,HOLD ;PREPARE TO GET SYNC CHAR PONTER
000548 000740 000017 BR ;PREPARE TO GET SYNC CHAR PONTER
000552 000740 000017 BIC #B10,R5 ;INSURE EVEN ADDRESS
000556 000740 000017 TST ;TO SYNC CHARS
000560 000740 177460 TST PING.PONG ;WHAT SET OF LINES?
000564 001401 177460 BEQ ;BR IF LOW SET
000568 022525 CMP (R5)+,(R5)+ ;SET OFFSET TO LAST TWO SYNC LINES.
```

```
382 ;
383 ;
384 ;
385 ;
386 ;
387 ;
388 ;
389 ;
390 ;
391 ;
392 ;
393 ;
394 ;
395 ;
396 ;
397 ;
398 ;
399 ;
400 ;
401 ;
402 ;
403 ;
404 ;
405 ;
406 ;
407 ;
408 ;
409 ;
410 ;
411 ;
412 ;
413 ;
414 ;
415 ;
416 ;
417 ;
418 ;
419 ;
420 ;
421 ;
422 ;
423 ;
424 ;
425 ;
426 ;
427 ;
428 ;
429 ;
430 ;
431 ;
432 ;
433 ;
434 ;
435 ;
436 ;
437 ;

000620 006267 177460 2$: ASR SELECT ;ACTIVE?
000624 103417 2$: RCS 4$ ;YES
000628 001002 2$: BNE 3$ ;BR IF NOT DONE
000630 000167 000556 3$: JMP #0,R0 ;JUMP IF DONE
000634 062701 000012 3$: ADD #12,R1 ;POP DVSCR POINTER
000638 010105 000001 3$: MOV R1,R5 ;PREPARE TO GET SYNC CHAR POINTER
000642 042705 000001 3$: BIC #B10,R5 ;INSURE EVEN ADDRESS
000646 062705 000040 177426 TST ;TO SYNC CHARS
000650 062705 000040 177426 ADD #32,HOLD ;POP BUFFER POINTER
000654 000756 000040 177426 BR ;CONTINUE
;
; ROUTINE USED TO CLEAR ALL DV11
; SECONDARY REGISTERS FOR ALL LINES
; THIS MAKES SURE NO JUNK IS LEFT FOR UCPU TO FIND.
4$: MOV #RRESSET,QDVSRCR ;INITIALIZE DV11
5$: CLR DVSR(R0) ;ZERO RAS
6$: ADD #C<BIT11+BIT10>+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+R10,DVSR(R0) ;ZERO RAS
7$: BNE ;UPDATE TO NEXT LINE AND SEC RFG BR IF NOT DONE
8$: MOV B,(R1),R3 ;*GET ACTIVE LINES
9$: CLR R2 ;ZERO LINE NO. IMAGE
10$: TST PING.PONG ;1ST GROUP OR SECOND?
11$: BEQ ;BR IF 1ST
12$: BIS #BIT3,R2 ;ADJUST LINES
13$: MOV #4,XCNT ;SET TO CHANGE SYNC CHARS EVERY 4 LINES
14$: MOV #1,17$ ;SET FOR PAD(JUNK) CHAR COUNT.
15$: CLC ;CLEAR CARRY
16$: RORB ;*LINE ACTIVE?
17$: BCS ;BR IF YES
18$: INC ;UPDATE LINE POINTER
19$: DEC XCNT ;4 LINES DONE?
20$: BNE ;BR IF NO
21$: MOV (R5)+,XCNT ;POP POINTER TO NEXT SYNCs
22$: TST ;RESET COUNTER
23$: MOV R3,XCNT ;*ALL LINES DONE?
24$: TSTB ;BR IF NO
25$: BNE ;CONT
26$: MOV #FUNC.X,R4 ;GET SECONDARY REGISTER DATA
27$: MOV R2,DVSR(R0) ;LOAD LINE NUMBER
28$: CLR DVSRSH(R0) ;CLEAR DVSRSH IMAGE
29$: BIS #BIT15+RENABL+MODE,4 ;WAIT FOR STROBE TO FINISH
30$: TST DVLCR(R0) ;*WAIT FOR STROBE TO FINISH
31$: BMI ;BR IF NOT DONE
32$: MOV (R4)+,DVSR(R0) ;LOAD SECONDARY REGISTERS
33$: MOV DVSRSH(R0) ;UPDATE SECONDARY REG POINTER
34$: TSTB ;ALL DONE?
35$: BNE ;BR IF NO
36$: MOV R2,DVSR(R0) ;LOAD LINE
37$: MOV R2,DVSRSH(R0) ;LOAD LINE
38$: MOV #5,VA ;SET ALTERNATE BA FOR EA BITS.
39$: JSR PC,SET.EA ;GO GET EA BITS.
40$: CLR STORE ;PREPARE TO
41$: MOV (R5),STORE ;CLEAR TABLE+SYNC+MODE
42$: ADD #CONTROL.TABLE,STORE ;STORE
43$: MOV #0,STORE ;SO THAT CNTRL BYTE IS =0
44$: CLR B ;IS THIS AN ASYNC LINE CARD?
45$: CMBP ;BR IF NOT ASYNC.
46$: BNE 18$
```

```
438 001112 005067 171174 CLR STORE ;#ZERO
439 001112 111567 171170 MOV# (R5) STORE ;#GET "SYNC" CHAR (S/B 377 FOR ASYNC)
440 001122 062767 002156 ADD #RX CONTROL TABLE STORE
441 001130 112777 000020 MOV# #BIT4 STORE ;#SET DSCARD AND MODE ZERO.
442 001136 012760 120000 MOV #BIT15+BIT13,DVLCR(R0) ;#SET RX ENABLE.
443 001144 005760 000004 TST DVLCR(R0) ;#WAIT FOR STROBE TO FINISH
444 001150 100775 -4 ;#WAIT
445 001152 012760 115000 MOV #BIT15+BIT12+BIT11+BIT9,DVLCR(R0) ;#STROBE IN 8 BITS/PER/CHAR
446 001160 005760 000004 TST DVLCR(R0) ;#WAIT FOR STROBE TO FINISH
447 001166 112760 172000 BMT -4 ;#WAIT FOR STROBE TO FINISH
448 001166 012760 172000 MOV #BIT15+BIT14+BIT13+BIT12+BIT10,DVLCR(R0) ;#STROBE IN 9600 BAUD RATE.
449 001174 005760 000004 TST DVLCR(R0) ;#WAIT FOR STROBE DONE.
450 001200 012760 107000 MOV #BIT15+BIT11+BIT10+BIT9,DVLCR(R0) ;#STROBE IN MAIN INTERNAL MODE
451 001202 012760 000004 TST DVLCR(R0) ;#WAIT FOR STROBE DONE
452 001210 005760 000004 BMT -4 ;#WAIT FOR STROBE DONE
453 001212 100775 000007 INCB DVSRSH(R0) ;SELECT BY CNT (ALTERNATE)
454 001222 105760 000010 MOV #2,DVSRAR(R0) ;SET FOR 2 SYNC CHARS
455 001230 112760 177756 MOV #2 STORE ;GET LINE NUMBER
456 001230 010267 177052 ASL STORE ;MULT BY 2 (MAKE EVEN)
457 001234 006367 177052 ADD #4,DVSRSH(R0) ;GET POINTER FOR RX BUFFER
458 001240 066767 177044 MOV# #4,DVSRSH(R0) ;SEL RX BA
459 001254 017767 177032 MOV# STORE,VA ;PREPARE TO GET EA BITS FOR RX BA.
460 001262 004767 000216 JSR PC,SET,EA ;GO GET THEM
461 001262 004767 000216 MOV# #6,DVSRSH(R0) ;SET TX TABLE BASE ADDR
462 001266 116067 176774 MOV# DVSRAR(R0),VA ;READ IT
463 001300 004767 000176 JSR PC,SET,EA ;SET EA BITS
464 001306 105260 000007 INCB DVSRSH(R0) ;SEL RX TABLE BASE ADDR
465 001312 016067 176756 MOV# PC,SET,EA ;READ IT
466 001320 004767 000160 JSR PC,SET,EA ;SET EA BITS
467 001324 105060 000007 CLRB DVSRSH(R0) ;SEL TX BA PRI.
468 001330 016067 000010 MOV# DVSRAR(R0),VA ;READ IT
469 001336 004767 000142 JSR PC,SET,EA ;SET EA BITS
470 001340 016767 176734 MOV# EA,PRI,EA ;SAVE PRINCIPLE EA BITS FOR ISR
471 001350 016767 176724 MOV# PA,PRI,PA ;SAVE PRINCIPLE PA FOR ISR
472 001356 062760 000020 ADD #20,DVSRAR(R0) ;POINT TXBA TO PAD(JUNK) CHARS.
473 001360 105260 000007 INCB DVSRSH(R0) ;SET TX BA BYTE CNT
474 001370 105260 000106 MOV# #17,DVSRAR(R0) ;SET NUMBER OF PAD(JUNK) CHARS TO BE SENT.
475 001376 005460 000010 NEG DVSRAR(R0) ;MAKE COUNT 2'S COMP.
476 001402 005267 000074 INC #5 ;PAD CNT=PAD CNT+1
477 001406 000167 177532 JMP B5 ;CNT
478 001410 016700 176370 MOV# ADDR,R0 ;GET DVSCR POINTER
479 001416 016767 176372 MOV# DVID1,SELECT ;GET ACTIVE DEVICES
480 001424 016701 176654 MOV# SELECT,R1 ;GET ACTIVE DVII'S
481 001430 006201 176654 ASR R1 ;ACTIVE?
482 001432 103404 176660 BCS 16$ ;BR IF YES
483 001434 001436 000040 BEQ 15$ ;BR IF DONE
484 001436 062700 000040 ADD #40,R0 ;POP DVSCR POINTER
485 001442 000772 177532 BR 14$ ;CNT
486 001444 012767 013146 176624 MOV# #TXBAS,VA ;GET EA BITS FOR ALTERNATE BA
487 001452 104415 000000 000276 GETPAS,BEGIN,VA ;GET PHYSICAL ADDRESS FROM 16-BIT VA
488 001460 116767 176616 011217 MOV# EA,ALT,EA ;GET EA BITS
489 001466 016767 176606 011214 MOV# PA,ALT,PA ;SAVE ALTERNATE EA BITS
490 001474 052710 030101 BIS #NPR.IE+STORE.IE,RX.IE+BIT0,@DVSCR ;SAVE ALTERNATE PA FOR ISR
491 001474 052710 030101 BIS #NPR.IE+STORE.IE,RX.IE+BIT0,@DVSCR ;SET MICO-PROCESSOR GO!!
492 001500 000756 BR 15$ ;CNT
```

```
494 001502 000000 17$: 0
495
496 001504 SET,EA:
497 001504 104415 000000 000276 GETPAS,BEGIN,VA ;GET PHYSICAL ADDRESS FROM 16-BIT VA
498 001512 042710 000060 BIC #BIT5+BIT4,@DVSCR ;CLEAR EA BITS
499
500 001516 056710 176560 BIS EA,@DVSCR ;SET EA BITS
501 001522 016760 176552 000010 MOV# PA,DVSRAR(R0) ;LOAD PA
502 001530 000207 RTS PC
```



```
504 -----  
505 BEGIN TO SCAN FOR ALL SELECTED LINES FOR  
506 ALL SELECTED DEVICES TO FINISH  
507 IT IS IMPORTANT TO NOTE THAT *ALL* SELECTED  
508 LINES FOR *ALL* SELECTED DEVICES MUST  
509 FINISH FOR AN END PASS CONSIDERATION.  
510 -----  
511  
512 SCANNER:  
513 MOV #100,R4 ;SET DELAY.  
514 CLR R5 ;SET FOR A LONG DELAY  
515  
516 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...  
517 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
518 TST ABORT ;SHOULD ROUTINE BE ABORTED?  
519 BR IF NO  
520 TST ABORT ;R1=TX ERR -1=RX ERR  
521 BMI 63$ ;BR IF RX ERROR  
522 ;***** TRANSMITTER ERROR! *****  
523 ;CHECK MADE IN THIS ORDER  
524 IF "CSRC" IS = TO DVSCR (SEL 0)  
525 THEN THE ERROR WAS: BIT 15 NOT SET ON INTERRUPT  
526 IF "CSRC" IS = TO DVNSR (SEL 14)  
527 THEN THE ERROR WAS: NEITHER PRINCIPAL OR  
528 ALTERNATE BYTE COUNT WAS THE ENTRY IN DVNSR  
529 MOV #20,ERRTYP ;UNKNOWN XMITTER ERROR  
530 ;*****  
531 ORDERS,BEGIN, NULL ;SEE ABOVE  
532 ;*****  
533 BR 64$ ;CONTINUE  
534 -----  
535 ;***** RECEIVER ERROR! *****  
536 ;CHECK MADE IN THIS ORDER  
537 IF "CSRC" IS = TO DVSCR (SEL 0)  
538 THEN THE ERROR WAS: BIT 07 NOT SET ON INTERRUPT  
539 IF "CSRC" IS = TO DVNSR (SEL 14)  
540 THEN THE ERROR WAS: TRANS UNDERRUN OCCURED  
541 IF "CSRC" IS = TO DVVIC (SEL 2)  
542 THEN THE ERROR WAS: BCC ERROR OR  
543 ILLEGAL ERROR CONDITION  
544  
545 001602* 012767 000017 176276 63$: MOV #17,ERRTYP ;UNKNOWN RECEIVER ERROR  
546 001602* 012767 000017 176276 ;*****  
547 001610* 104405 000000* 000000 ORDERS,BEGIN, NULL ;SEE ABOVE  
548 001610* 104405 000000* 000000 ;*****  
549 001616* 000167 176520 64$: JMP RESTRT ;RESTART MODULE  
550
```

```
551 001622* 026767 176376 015174 65$: CMP LINES1,XXX1 ;DID ALL LINES FOR 1ST DV11 FINISH?  
552 001630* 001113 BNE IOS ;BR IF NO  
553 001640* 176400 015204 CMP LINES2,XXX2 ;DID ALL LINES FOR 2ND DV11 FINISH?  
554 001640* 011107 BNE IOS ;BR IF NO  
555 001642* 026767 176402 015214 CMP LINES3,XXX3 ;DID ALL LINES FOR 3RD DV11 FINISH?  
556 001650* 011103 BNE IOS ;BR IF NO  
557 001657* 026767 176404 015224 CMP LINES4,XXX4 ;DID ALL LINES FOR THE 4TH FINISH?  
558 001660* 011077 BNE IOS ;BR IF NO  
559 001662* 016767 176120 176210 MOV ADDR,CSRA ;LOAD BASE DV11 CSR  
560 001670* 012767 017106* 000206 MOV #BUFFER.TABLE,BUFFPNT ;BUFFER TABLE, BUFFPNT  
561  
562 001676* 012701 000224* MOV #LINES1,R1 ;SET POINTER OF ALL RECV BUFFERS.  
563 001702* 005767 176366* TST PING.PONG ;WHAT SET?  
564 001706* 001404 BEQ 2$ ;JL  
565 001716* 005201 INC #16,BUFFPNT ;HIGH SET  
566 001720* 111102 MOVVB (R1),R2 ;GET ACTIVE LINES INTO R2  
567 001726* 016700 MOVVB BUFFPNT,R0 ;SET POINTER  
568 001726* 006267 176352* ASR SELECT ;SPCOL FOR ACTIVE DEVICES  
569 001732* 103417 BCS 4$ ;BR IF THE DEVICE WAS ACTIVE  
570 001734* 001417 BEQ 5$ ;BR IF ALL DEVICES DONE  
571 001736* 062767 000040 176134 99$: ADD #40,CSRA ;UPDATE CSR IMAGE  
572 001742* 062767 000040 000132 3$: ADD #32,BUFFPNT ;POP AN ENTIRE DV11S LINE BUFFER  
573 001752* 062701 000012 BR #2,R1 ;POP TO NEXT GROUP OF LINES  
574 001756* 000760 BR 2$ ;CONTINUE ALONG  
575 001760* 000241 CLC ;CLEAR CPU CARRY  
576 001762* 106002 RORB R2 ;SPCOL FOR ACTIVE LINES  
577 001764* 103411 BCS 7$ ;BR IF LINE WAS ACTIVE  
578 001766* 001763 BEQ 99$ ;BR IF ALL LINES FOR DV11 DONE  
579 001770* 005720 TST (R0)+ ;POP BUFFER POINTER TO NEXT BUFFER  
580 001772* 000772 BR 4$ ;CONTINUE ALONG  
581 001774* 005167 176274 5$: CDM PING.PONG ;PREPARE SELECTOR  
582 002000* 104413 000000* ENDTLS,BEGIN ;SIGNAL END OF ITERATION.  
583 ;MONITOR SHALL TEST END OF PASS  
584 -----  
585 002004* 000167 176332 6$: JMP RESTRT ;RESTART PROGRAM  
586 002010* 012004 013012* 7$: MOV (R0)+,R4 ;GET RECEIVER BUFFER POINTER  
587 002012* 012705 MOV #DATA.TABLE,R5 ;GET GOOD DATA POINTER  
588 002016* 121514 8$: CMPB (R5),(R4) ;DO THE DATA CHECKING  
589 002020* 001412 BEQ 9$ ;BR IF DATA IS GOOD  
590 002022* 010567 176054 MOV R5,SBADR ;PREPARE TO REPORT DATA ERROR!  
591 002024* 010467 176052 MOV R4,SABDR ;LOAD GOOD ADDR AND BAD ADDR  
592 002026* 111567 176050 MOVVB (R5),ASB ;LOAD GOOD DATA  
593 002036* 111467 176046 MOVVB (R4),AWAS ;LOAD BAD DATA  
594 ;*****  
595 002042* 104404 000000* 9$: CLRB (R4)+ ;DATA ERROR!!!  
596 ;*****  
597 002046* 105024 000076 9$: CMPB #76,(R5)+ ;CLEAR BUFFER. MAY BE PATCHED TO "TSTB (R4)+"(105724)  
598 002050* 012725 BNE 4$ ;HAS ALL DATA BEEN CHECKED?  
599 002052* 001360 BR 4$ ;BR NO  
600 002056* 000740 BR 10$ ;DON'T ALL DATA CHECKED.  
601 002060* 005305 10$: DEC R5 ;STALL FOR ALL DV11'S TO FINISH  
602 002062* 005304 BNE 1$ ;BR IF DELAY NOT =0  
603 002064* 005304 DEC R4 ;DELAY COUNT  
604 002066* 001223 BNE 11$ ;KEEP WAITING.  
605
```

```
607 002070* 104403 000000* 002106* MSGNS,BEGIN,HUNG ;ASCII MESSAGE CALL WITH COMMON HEADER
608 002076* 104410 000000* ENDS,BEGIN ;
609 002102* 000000* ABORT: .WORD 0 ;ABORT FLAG
610 002104* 000000* BUFPNT: 0 ;
611 002106* 002112* HUNG: .XWORD 0 ;
612 002110* 177777* HUNG: .XWORD 0 ;
613 002112* 042045 030526 020061 XHUNG: .ASCIZ "ADV11 MODULE IS HUNG. SEE LISTING3" ;
614 002120* 047515 052504 042514 ;
615 002126* 044440 020123 052510 ;
616 002134* 043516 020056 042523 ;
617 002142* 020105 044514 052123 ;
618 002150* 047111 022507 000 ;
619 002156* .EVEN
```

```
620 002156* RX.CONTROL.TABLE:
621 ;UNUSED AREA OF CONTROL TABLE IS FILLED WITH "GEN INT"(MODE0)
622 ;TO CATCH RECEIVER GOING TO WRONG CNTRL BYTE.
623
624
625 002156*
626 002156*
627 002156* 010
628 002157* 010
629 002160* 010
630 002161* 010
631 002162* 010
632 002163* 010
633 002164* 010
634 002165* 070
635 002166* 020
636 002167* 020
637 002170* 020
638 002171* 020
639 002172* 020
640 002173* 020
641 002174* 020
642 002175* 020
643 002176* 020
644 002177* 020
645 002200* 020
646 002201* 020
647 002202* 020
648 002203* 020
649 002204* 020
650 002205* 020
651
        .-RX.CONTROL.TABLE+0 ;MODE 0
        .-+0
        .-BYTE BIT3+MODE0 ;00 INC/RCC+MODE0
        .-BYTE BIT3+MODE0 ;01 INC/RCC+MODE0
        .-BYTE BIT3+MODE0 ;02 INC/RCC+MODE0
        .-BYTE BIT3+MODE0 ;03 INC/RCC+MODE0
        .-BYTE BIT3+MODE0 ;04 INC/RCC+MODE0
        .-BYTE BIT3+MODE0 ;05 INC/RCC+MODE0
        .-BYTE BIT3+MODE0 ;06 INC/RCC+MODE0
        .-BYTE BIT4+BIT3+MODE1 ;07 DSCARD+INC/RCC+MODE1
        .-BYTE BIT4+MODE0 ;10 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;11 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;12 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;13 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;14 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;15 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;16 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;17 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;20 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;21 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;22 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;23 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;24 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;25 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;26 DSCARD+MODE0
        .-BYTE BIT4+MODE0 ;27 DSCARD+MODE0
```

```
652 002556*
653 002556*
654 002566*
655 002567*
656 002570*
657 002571*
658 002574*
659 002574*
660 002574*
661 002574*
662 002574*
663 002574*
664 002575*
665 003156*
666 003176*
667 003177*
668 003200*
669 003201*
670 003202*
671 003203*
672 003204*
673 003205*
674 003506*
675 003606*
676 003606*
677 003607*
678 003610*
679 003611*
680 003612*
681 003613*
682 003614*
683 003615*
684

.=RX.CONTROL.TABLE+400 ;MODE1
.=.+10
  .BYTE BIT3+MODE1 ;10 INC/BCC+MODE1
  .BYTE BIT3+MODE1 ;11 INC/BCC+MODE1
  .BYTE BIT3+MODE1 ;12 INC/BCC+MODE1
  .BYTE BIT3+MODE1 ;13 INC/BCC+MODE1
  .BYTE BIT3+MODE1 ;14 INC/BCC+MODE1
  .BYTE BIT3+MODE1 ;15 INC/BCC+MODE1
  .BYTE BIT3+MODE1 ;16 INC/BCC+MODE1
  .BYTE BIT4+BIT3+MODE2 ;17 DSCARD+INC/BCC+MODE2

.=RX.CONTROL.TABLE+1000 ;MODE2
.=.+20
  .BYTE BIT3+BIT1+MODE2 ;20 INC/BCC+STORE+MODE2
  .BYTE BIT4+BIT3+MODE2 ;21 DSCARD+INC/BCC+MODE2
  .BYTE BIT3+BIT1+MODE2 ;22 INC/BCC+STORE+MODE2
  .BYTE BIT4+BIT3+MODE2 ;23 DSCARD+INC/BCC+MODE2
  .BYTE BIT3+BIT1+MODE2 ;24 INC/BCC+STORE+MODE2
  .BYTE BIT4+BIT3+MODE2 ;25 DSCARD+INC/BCC+MODE2
  .BYTE BIT3+BIT1+MODE2 ;26 INC/BCC+STORE+MODE2
  .BYTE BIT4+BIT3+MODE3 ;27 DSCARD+INC/BCC+MODE3

.=RX.CONTROL.TABLE+1400 ;MODE3
.=.+30
  .BYTE BIT1+MODE3 ;30 RESERVED+MODE3
  .BYTE BIT1+MODE3 ;31 RESERVED+MODE3
  .BYTE BIT1+MODE3 ;32 RESERVED+MODE3
  .BYTE BIT1+MODE3 ;33 RESERVED+MODE3
  .BYTE BIT1+MODE3 ;34 RESERVED+MODE3
  .BYTE BIT1+MODE3 ;35 RESERVED+MODE3
  .BYTE BIT1+MODE3 ;36 RESERVED+MODE3
  .BYTE BIT1+MODE4 ;37 RESERVED+MODE4
```

```
685 004156*
686 004216*
687 004217*
688 004218*
689 004220*
690 004221*
691 004222*
692 004223*
693 004224*
694 004225*
695
696 004556*
697 004626*
698 004627*
699 004630*
700 004631*
701 004632*
702 004633*
703 004634*
704 004635*
705 005103*
706 005233*
707 005236*
708 005237*
709 005240*
710 005241*
711 005242*
712 005243*
713 005244*
714 005245*
715 005245*
716
717 005556*
718 005646*
719 005647*
720 005648*
721 005649*
722 005650*
723 005651*
724 005652*
725 005653*
726 005654*
727 005655*
728

.=RX.CONTROL.TABLE+2000 ;MODE4
.=.+40
  .BYTE BIT3+BIT1+MODE4 ;40 INC/BCC+RESERVED+MODE4
  .BYTE BIT3+BIT1+MODE4 ;41 INC/BCC+RESERVED+MODE4
  .BYTE BIT3+BIT1+MODE4 ;42 INC/BCC+RESERVED+MODE4
  .BYTE BIT3+BIT1+MODE4 ;43 INC/BCC+RESERVED+MODE4
  .BYTE BIT3+BIT1+MODE4 ;44 INC/BCC+RESERVED+MODE4
  .BYTE BIT3+BIT1+MODE4 ;45 INC/BCC+RESERVED+MODE4
  .BYTE BIT3+BIT1+MODE4 ;46 INC/BCC+RESERVED+MODE4
  .BYTE BIT1+MODE5 ;47 RESERVED+MODE5

.=RX.CONTROL.TABLE+2400 ;MODE5
.=.+50
  .BYTE BIT3+BIT1+MODE5 ;50 INC/BCC+RESERVED+MODE5
  .BYTE BIT3+BIT1+MODE5 ;51 INC/BCC+RESERVED+MODE5
  .BYTE BIT3+BIT1+MODE5 ;52 INC/BCC+RESERVED+MODE5
  .BYTE BIT3+BIT1+MODE5 ;53 INC/BCC+RESERVED+MODE5
  .BYTE BIT3+BIT1+MODE5 ;54 INC/BCC+RESERVED+MODE5
  .BYTE BIT3+BIT1+MODE5 ;55 INC/BCC+RESERVED+MODE5
  .BYTE BIT3+BIT1+MODE5 ;56 INC/BCC+RESERVED+MODE5
  .BYTE BIT1+MODE6 ;57 RESERVED+MODE6

.=RX.CONTROL.TABLE+3000 ;MODE6
.=.+60
  .BYTE BIT3+BIT1+MODE6 ;60 INC/BCC+STORE+MODE6
  .BYTE BIT4+BIT3+MODE6 ;61 DSCARD+INC/BCC+MODE6
  .BYTE BIT3+BIT1+MODE6 ;62 INC/BCC+STORE+MODE6
  .BYTE BIT4+BIT3+MODE6 ;63 DSCARD+INC/BCC+MODE6
  .BYTE BIT3+BIT1+MODE6 ;64 INC/BCC+STORE+MODE6
  .BYTE BIT4+BIT3+MODE6 ;65 DSCARD+INC/BCC+MODE6
  .BYTE BIT3+BIT1+MODE6 ;66 INC/BCC+STORE+MODE6
  .BYTE BIT4+BIT3+MODE7 ;67 DSCARD+INC/BCC+MODE7

.=RX.CONTROL.TABLE+3400 ;MODE7
.=.+70
  .BYTE BIT3+BIT1+MODE7 ;70 INC/BCC+RESERVED+MODE7
  .BYTE BIT3+BIT1+MODE7 ;71 INC/BCC+RESERVED+MODE7
  .BYTE BIT3+BIT1+MODE7 ;72 INC/BCC+RESERVED+MODE7
  .BYTE BIT3+BIT1+MODE7 ;73 INC/BCC+RESERVED+MODE7
  .BYTE BIT3+BIT1+MODE7 ;74 INC/BCC+RESERVED+MODE7
  .BYTE BIT3+BIT1+MODE7 ;75 INC/BCC+RESERVED+MODE7
  .BYTE BIT3+BIT2+MODE7 ;76 INC/BCC+EXP/BCC+MODE7
  .BYTE BIT0+MODE0 ;77 GEN/INT+MODE0
```

```
729 006156* 006156*
730
731
732 006156* 006156*
733 006156* 006156*
734 006156* 010
735 006157* 010
736 006160* 010
737 006161* 010
738 006162* 010
739 006163* 010
740 006164* 010
741 006165* 050
742 006166* 001
743 006167* 001
744 006170* 001
745 006171* 001
746 006172* 001
747 006173* 001
748 006174* 001
749 006175* 001
750 006176* 001
751 006177* 001
752 006200* 001
753 006201* 001
754 006202* 001
755 006203* 001
756 006204* 001
757 006205* 001
758 006206* 000074
759 006207* 000074
760 006376* 000074
761 006472* 000074
762
```

```
.=RX.CONTROL.TABLE+4000
TX.CONTROL.TABLE:
.=TX.CONTROL.TABLE+0 ;MODE 0
=-.+0
.BYTE BIT3+MODE0 ;00 INC/BCC+MODE0
.BYTE BIT3+MODE0 ;01 INC/BCC+MODE0
.BYTE BIT3+MODE0 ;02 INC/BCC+MODE0
.BYTE BIT3+MODE0 ;03 INC/BCC+MODE0
.BYTE BIT3+MODE0 ;04 INC/BCC+MODE0
.BYTE BIT3+MODE0 ;05 INC/BCC+MODE0
.BYTE BIT3+MODE0 ;06 INC/BCC+MODE0
.BYTE BIT3+MODE0 ;07 INC/BCC+MODE0
.BYTE BIT0+MODE0 ;10 RESERVED+MODE0
.BYTE BIT0+MODE0 ;11 RESERVED+MODE0
.BYTE BIT0+MODE0 ;12 RESERVED+MODE0
.BYTE BIT0+MODE0 ;13 RESERVED+MODE0
.BYTE BIT0+MODE0 ;14 RESERVED+MODE0
.BYTE BIT0+MODE0 ;15 RESERVED+MODE0
.BYTE BIT0+MODE0 ;16 RESERVED+MODE0
.BYTE BIT0+MODE0 ;17 RESERVED+MODE0
.BYTE BIT0+MODE0 ;20 RESERVED+MODE0
.BYTE BIT0+MODE0 ;21 RESERVED+MODE0
.BYTE BIT0+MODE0 ;22 RESERVED+MODE0
.BYTE BIT0+MODE0 ;23 RESERVED+MODE0
.BYTE BIT0+MODE0 ;24 RESERVED+MODE0
.BYTE BIT0+MODE0 ;25 RESERVED+MODE0
.BYTE BIT0+MODE0 ;26 RESERVED+MODE0
.BYTE BIT0+MODE0 ;27 RESERVED+MODE0
RXB10: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #0
RXB11: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #1
RXB12: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #2
RXB13: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #3
.MEXIT
```

```
763 006556* 006556*
764 006556* 006556*
765 006566* 050
766 006567* 050
767 006570* 050
768 006571* 050
769 006572* 050
770 006573* 050
771 006574* 050
772 006575* 110
773 006576* 000074
774 006672* 000074
775 006766* 000074
776 007062* 000074
777
778
779 007156* 007156*
780 007176* 111
781 007177* 111
782 007178* 111
783 007200* 111
784 007201* 111
785 007202* 111
786 007203* 111
787 007204* 111
788 007205* 151
789 007206* 000074
790 007302* 000074
791 007303* 000074
792 007376* 000074
793 007472* 000074
```

```
.=TX.CONTROL.TABLE+400 ;MODE1
=-.+10
.BYTE BIT3+MODE1 ;10 INC/BCC+MODE1
.BYTE BIT3+MODE1 ;11 INC/BCC+MODE1
.BYTE BIT3+MODE1 ;12 INC/BCC+MODE1
.BYTE BIT3+MODE1 ;13 INC/BCC+MODE1
.BYTE BIT3+MODE1 ;14 INC/BCC+MODE1
.BYTE BIT3+MODE1 ;15 INC/BCC+MODE1
.BYTE BIT3+MODE1 ;16 INC/BCC+MODE1
.BYTE BIT3+MODE2 ;17 INC/BCC+MODE2
RXB14: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #4
RXB15: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #5
RXB16: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #6
RXB17: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #7
.MEXIT
.=TX.CONTROL.TABLE+1000 ;MODE2
=-.+20
.BYTE BIT3+BIT0+MODE2 ;20 INC/BCC+RESERVED+MODE2
.BYTE BIT3+BIT0+MODE2 ;21 INC/BCC+RESERVED+MODE2
.BYTE BIT3+BIT0+MODE2 ;22 INC/BCC+RESERVED+MODE2
.BYTE BIT3+BIT0+MODE2 ;23 INC/BCC+RESERVED+MODE2
.BYTE BIT3+BIT0+MODE2 ;24 INC/BCC+RESERVED+MODE2
.BYTE BIT3+BIT0+MODE2 ;25 INC/BCC+RESERVED+MODE2
.BYTE BIT3+BIT0+MODE2 ;26 INC/BCC+RESERVED+MODE2
.BYTE BIT3+BIT0+MODE3 ;27 INC/BCC+RESERVED+MODE3
RXB110: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #10
RXB111: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #11
RXB112: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #12
RXB113: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #13
.MEXIT
```

```
794          007556*  
795          007806*  
796 007606* 142  
797 007607* 142  
798 007610* 142  
799 007611* 142  
800 007612* 142  
801 007613* 142  
802 007614* 141  
803 007615* 202  
804 007616* 000074  
805 007712* 000074  
806 010806* 000074  
807 010102* 000074  
808  
809  
810  
811          010156*  
812          010216*  
813 010216* 211  
814 010217* 211  
815 010220* 211  
816 010221* 211  
817 010222* 211  
818 010223* 211  
819 010224* 211  
820 010225* 241  
821 010226* 000074  
822 010633* 000074  
823 010416* 000074  
824 010512* 000074  
825  
826          010556*  
827          010626*  
828 010626* 251  
829 010627* 251  
830 010630* 251  
831 010631* 251  
832 010632* 251  
833 010633* 251  
834 010634* 251  
835 010635* 251  
836 010635* 301  
837 010636* 000074  
838 010732* 000074  
839 011026* 000074  
840 011122* 000074  
841  
  
      -=TX_CONTROL.TABLE+1400      ;MODE3  
      -=.430  
      .BYTE BIT1+MODE3      ;30  SND/DLE+MODE3  
      .BYTE BIT1+MODE3      ;31  SND/DLE+MODE3  
      .BYTE BIT1+MODE3      ;32  SND/DLE+MODE3  
      .BYTE BIT1+MODE3      ;33  SND/DLE+MODE3  
      .BYTE BIT1+MODE3      ;34  SND/DLE+MODE3  
      .BYTE BIT1+MODE3      ;35  SND/DLE+MODE3  
      .BYTE BIT0+MODE3      ;36  RESERVED+MODE3  
      .BYTE BIT1+MODE4      ;37  SND/DLE+MODE4  
RXB114: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #14  
RXB115: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #15  
RXB116: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #16  
RXB117: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #1 LINE #17  
      .MEXIT  
  
      -=TX_CONTROL.TABLE+2000      ;MODE4  
      -=.440  
      .BYTE BIT3+BIT0+MODE4  ;40  INC/BCC+RESERVED+MODE4  
      .BYTE BIT3+BIT0+MODE4  ;41  INC/BCC+RESERVED+MODE4  
      .BYTE BIT3+BIT0+MODE4  ;42  INC/BCC+RESERVED+MODE4  
      .BYTE BIT3+BIT0+MODE4  ;43  INC/BCC+RESERVED+MODE4  
      .BYTE BIT3+BIT0+MODE4  ;44  INC/BCC+RESERVED+MODE4  
      .BYTE BIT3+BIT0+MODE4  ;45  INC/BCC+RESERVED+MODE4  
      .BYTE BIT3+BIT0+MODE4  ;46  INC/BCC+RESERVED+MODE4  
      .BYTE BIT0+MODE5      ;47  RESERVED+MODE5  
RXB20:  .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #0  
RXB21:  .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #1  
RXB22:  .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #2  
RXB23:  .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #3  
      .MEXIT  
  
      -=TX_CONTROL.TABLE+2400      ;MODE5  
      -=.450  
      .BYTE BIT3+BIT0+MODE5  ;50  INC/BCC+RESERVED+MODE5  
      .BYTE BIT3+BIT0+MODE5  ;51  INC/BCC+RESERVED+MODE5  
      .BYTE BIT3+BIT0+MODE5  ;52  INC/BCC+RESERVED+MODE5  
      .BYTE BIT3+BIT0+MODE5  ;53  INC/BCC+RESERVED+MODE5  
      .BYTE BIT3+BIT0+MODE5  ;54  INC/BCC+RESERVED+MODE5  
      .BYTE BIT3+BIT0+MODE5  ;55  INC/BCC+RESERVED+MODE5  
      .BYTE BIT3+BIT0+MODE5  ;56  INC/BCC+RESERVED+MODE5  
      .BYTE BIT0+MODE6      ;57  RESERVED+MODE6  
RXB24:  .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #4  
RXB25:  .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #5  
RXB26:  .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #6  
RXB27:  .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #7  
      .MEXIT
```

```
842          011156*  
843          011236*  
844 011236* 311  
845 011240* 311  
846 011241* 311  
847 011241* 311  
848 011242* 311  
849 011243* 311  
850 011244* 311  
851 011245* 351  
852 011246* 000074  
853 011342* 000074  
854 011342* 000074  
855 011532* 000074  
856  
857          011556*  
858          011646*  
859 011646* 351  
860 011647* 351  
861 011647* 351  
862 011650* 351  
863 011651* 351  
864 011652* 351  
865 011653* 351  
866 011654* 354  
867 011655* 001  
868 011656* 000074  
869 011752* 000074  
870 012046* 000074  
871 012142* 000074  
872  
873  
  
      -=TX_CONTROL.TABLE+3000      ;MODE6  
      -=.460  
      .BYTE BIT3+BIT0+MODE6  ;60  INC/BCC+RESERVED+MODE6  
      .BYTE BIT3+BIT0+MODE6  ;61  INC/BCC+RESERVED+MODE6  
      .BYTE BIT3+BIT0+MODE6  ;62  INC/BCC+RESERVED+MODE6  
      .BYTE BIT3+BIT0+MODE6  ;63  INC/BCC+RESERVED+MODE6  
      .BYTE BIT3+BIT0+MODE6  ;64  INC/BCC+RESERVED+MODE6  
      .BYTE BIT3+BIT0+MODE6  ;65  INC/BCC+RESERVED+MODE6  
      .BYTE BIT3+BIT0+MODE6  ;66  INC/BCC+RESERVED+MODE6  
      .BYTE BIT3+BIT0+MODE6  ;67  INC/BCC+RESERVED+MODE6  
RXB210: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #10  
RXB211: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #11  
RXB212: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #12  
RXB213: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #13  
      .MEXIT  
  
      -=TX_CONTROL.TABLE+3400      ;MODE7  
      -=.470  
      .BYTE BIT3+BIT0+MODE7  ;70  INC/BCC+RESERVED+MODE7  
      .BYTE BIT3+BIT0+MODE7  ;71  INC/BCC+RESERVED+MODE7  
      .BYTE BIT3+BIT0+MODE7  ;72  INC/BCC+RESERVED+MODE7  
      .BYTE BIT3+BIT0+MODE7  ;73  INC/BCC+RESERVED+MODE7  
      .BYTE BIT3+BIT0+MODE7  ;74  INC/BCC+RESERVED+MODE7  
      .BYTE BIT3+BIT0+MODE7  ;75  INC/BCC+RESERVED+MODE7  
      .BYTE BIT3+BIT0+MODE7  ;76  INC/BCC+RESERVED+MODE7  
      .BYTE BIT0+MODE8      ;77  RESERVED+MODE8  
RXB214: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #14  
RXB215: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #15  
RXB216: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #16  
RXB217: .BLKB 74      ;THIS IS RECEIVER BUFFER FOR DEVICE #2 LINE #17  
      .MEXIT
```

874	012236*								
875	012236*	010046							
876	012240*	010446							
877	012244*	032710							
878	012244*	032710	002000						
879	012250*	001002							
880	012252*	005710							
881	012254*	100407							
882	012256*	010067	165616						
883	012262*	011067	165616						
884	012266*	010067	165610						
885	012272*	000447							
886	012274*	016004	000014						
887	012300*	110405							
888	012302*	042705	177400						
889	012306*	052705	100400						
890	012312*	026504							
891	012314*	001416							
892	012316*	052705	001000						
893	012322*	020504							
894	012324*	001412							
895	012326*	010067	165546						
896	012332*	010067	165544						
897	012344*	010467	000014						
898	012344*	010467	165534						
899	012350*	000420							
900	012352*	010460	000006						
901	012362*	027710	177400						
902	012364*	042710	000060						
903									
904	012370*	105360	000007						
905	012374*	010114							
906	012376*	156710	000302						
907	012402*	016760	000300	000010					
908	012410*	000413							
909	012412*	027710	004000						
910	012416*	012767	000001	167456					
911	012424*	000405							
912	012426*	156710	000253						
913	012444*	016760	000252	000010					
914	012440*								
915	012440*	012604							
916	012442*	012600							
917	012444*	016760							
918	012446*	000002							

```

TXISR:
MOV R0,-(SP) ;SAVE R0 ON THE STACK
MOV R4,-(SP) ;SAVE R4 ON THE STACK
MOV (R5),R0 ;GET DEVICE CSR ON INTERRUPT
RTI #QVRFLO,@DVSCR ;OVERFLOW
BNE 25 ;REPORT OVERFLOW
1$: TST @DVSCR ;IS 15=1?
BWI 35 ;BR IF YES
2$: MOV R0,CSRA ;LOAD FOR PRINT OUT
MOV @DVSCR,ASTAT ;GET BAD LOC SET UP
MOV R0,ACSR ;SET FOR TYPE OUT
3$: BR 65$ ;LEAVE ROUTINE
MOV DVNSR(R0),R4 ;GET DVNSR REGISTER
MOVB R4,R5 ;GET LINE NUMBER
BIC #<<377>,R5 ;CLEAR ANY SIGN EXTEND
BIS #BIT15+BIT8,R5 ;SET 1 PROBABILITY OF GOOD RESULTS
CMP R5,R4 ;ARE THEY EQUAL?
BEQ 45 ;BR IF YES
BIS #BIT9,R5 ;GET 2ND CHOICE OF GOOD RESULTS
CMP R5,R4 ;NOW ARE THESE OK?
BEQ 45 ;BR IF YES
MOV R0,CSRA ;SAVE FOR PRINTOUT
MOV R0,ACSR ;SET FOR ERROR
MOV #14,ACSR ;POINTER TO DVNSR
MOV R4,ASTAT ;GET BAD RESULTS
BR 65$ ;LEAVE ROUTINE
4$: MOV R4,DVSR5(R0) ;LOAD LINE NUMBER
MOV #40,DVSR4(R0) ;SET FOR 40(B) CHARS
BIC #BIT5+BIT4,@DVSCR ;CLEAR EA BITS
DECB DVSR5H(R0) ;IS THIS PRI OR ALT?
BNE 5 ;BR IF IT WAS ALT
BISB PRI,EA,@DVSCR ;SET EA BITS
MOV PRI,PA,DVSR4(R0) ;LOAD PRI
BR 65 ;GET OUT
5$: BIS #RESET,@DVSCR ;RESET DEVICE ON ERROR
MOV #1,ABORT ;SET ERROR FLAG
BR 65
6$: ALT,EA,@DVSCR ;SET EA BITS
ALT,PA,DVSR4(R0) ;LOAD ALTERNATE
MOV (SP)+,R4 ;POP STACK TO R4
MOV (SP)+,R0 ;POP STACK TO R0
MOV (SP)+,R5 ;POP STACK TO R5
RTI ;LEAVE ISR
  
```

919	012450*								
920	012450*	010046							
921	012454*	010446							
922	012454*	032710							
923	012456*	052710	000000						
924	012462*	016001	000002						
925	012466*	107710							
926	012470*	100407							
927	012472*	010067	165402						
928	012476*	011067	165402						
929	012502*	010067	165374						
930	012506*	000462							
931	012510*	032701	170000						
932	012514*	001037							
933	012516*	122701							
934	012522*	001043	000077						
935	012522*	001043							
936	012524*	000301							
937	012526*	042710	177400						
938	012532*	010160	000006						
939	012536*	112760	000013	000007					
940	012544*	042760	000004	000010					
941	012552*	052760	000002	000010					
942	012560*	112760	000017	000007					
943	012566*	052760	000020	000010					
944	012574*	052710	000400						
945	012600*	062705	000010						
946	012604*	006301							
947	012606*	056115	012712*						
948	012612*	000430							
949	012614*	010110							
950	012616*	042705	170377						
951	012622*	052705	050000						
952	012626*	020501							
953	012630*	001417							
954	012632*	010067	165244						
955	012636*	062767	000002	165236					
956	012644*	010167	165234						
957	012650*	010067	165224						
958	012654*	012710	004000						
959	012660*	012767	177777	167214					
960	012666*	000402							
961	012672*	052710	000400						
962	012674*								
963	012674*	012601							
964	012676*	012600							
965	012700*	012605							
966	012702*	000002							

```

RXISR:
MOV R0,-(SP) ;SAVE R0 ON THE STACK
MOV R1,(SP) ;SAVE R1 ON THE STACK
MOV (R5),R0 ;GET DVSCR FOR ISR USE
BIS #0,@DVSCR ;NOP- (DATAIP BANG DV11 REGISTER)
MOV DVRC(R0),R1 ;GET DVRC REGISTER
TSTB @DVSCR ;DVSCR BIT7=1?
BWI 15 ;BR IF YES
MOV R0,CSRA ;SAVE FOR TYPE OUT
MOV @DVSCR,ASTAT ;GET BAD DATA
MOV R0,ACSR ;BR IF ERROR
BIT #BIT15+BIT14+BIT13+BIT12,R1 ;ARE ANY "ERROR CODES" SET?
BNE #7,R1 ;BR IF YES (BCC S/R THE ONLY)
2$: SWAB R1 ;GET LINE IN LOW BYTE
BIC #<377+400>,R1 ;CLEAR HIGH BYTE
MOV R1,DVSR5(R0) ;LOAD LINE NUMBER
MOVB #15,DVSR5H(R0) ;SEL LINE STATE REGISTER
BIC #TX.GO,DVSPA(R0) ;CLEAR TX GC
BIS #RESYNC,DVSR4(R0) ;SET RX RESYNC
MOVB #17,DVSR5H(R0) ;SEL CTRL BYTE STORE
BIS #DISCARD,DVSR4(R0) ;THROW AWAY THE 77 CHAR
BIS #BIT8,@DVSCR ;RESTART DV11 UCPU
ADD R1,R5 ;POP TO END PASS FLAG POINTER
ROR R1,R5 ;MAKE EVEN (MULT BY 2)
BIS EOP,TABLE(R1),(R5) ;SET END-OF-PASS FLAG.
BR 95 ;CONTINUE ALONG
5$: MOV R1,R5 ;LOAD LINE NUMBER INTO R5
BIC #<<BIT11+BIT10+BIT9+BIT8>,R5 ;MAKE IT EQUAL TO ECC INDICATOR
BIS #BIT14+BIT12,R5 ;MAKE IT EQUAL TO ECC INDICATOR
CMP R5,R1 ;IS RIC OK?
BEQ 95 ;BR IF YES
6$: MOV R0,ACSR ;SET FOR ERROR
ADD #2,ACSR ;POINT TO DVFC
MOV R1,ASTAT ;LOAD BAD RESULTS
MOV R0,CSRA ;SET FOR PRINT OUT
7$: #RESET,@DVSCR ;SET ERROR INDICATOR.
MOV #1,ABORT ;SET ERROR INDICATOR.
BR 95 ;CONT
8$: BIS #BIT8,@DVSCR ;RESTART DV11
9$: MOV (SP)+,R1 ;POP STACK TO R1
MOV (SP)+,R0 ;POP STACK TO R0
MOV (SP)+,R5 ;POP STACK TO R5
RTI ;LEAVE ISR
  
```

```

967 012704* 001
968 012705* 001
969 012706* 000001
970 012710* 000001
971
972 012712* 000001
973 012713* 000002
974 012714* 000002
975 012715* 000002
976 012716* 000002
977 012717* 000002
978 012718* 000002
979 012719* 000002
980 012720* 000002
981 012721* 000002
982 012722* 000002
983 012723* 000002
984 012724* 000002
985 012725* 000002
986 012726* 000002
987 012727* 000002
988 012728* 000002
989 012729* 000002
990 012730* 000002
991 012731* 000002
992 012732* 000002
993 012733* 000002
994 012734* 000002
995 012735* 000002
996 012736* 000002
997 012737* 000002
998 012738* 000002
999 012739* 000002
1000 012740* 000002
1001 012741* 000002
1002 012742* 000002
1003 012743* 000002
1004 012744* 000002
1005 012745* 000002
1006 012746* 000002
1007 012747* 000002
1008 012748* 000002
1009 012749* 000002
1010 012750* 000002
1011 012751* 000002
1012 012752* 000002

```

```

PRI-FA: -BYTE 1
ALT-FA: -BYTE 1
PRI-PA: -WORD 1
ALT-PA: -WORD 1

```

```

EOP.TABLE:
B<000000000000000000000000> }LINE 0
B<000000000000000000000000> }LINE 1
B<000000000000000000000000> }LINE 2
B<000000000000000000000000> }LINE 3
B<000000000000000000000000> }LINE 4
B<000000000000000000000000> }LINE 5
B<000000000000000000000000> }LINE 6
B<000000000000000000000000> }LINE 7
B<000000000000000000000000> }LINE 8
B<000000000000000000000000> }LINE 9
B<000000000000000000000000> }LINE 10
B<000000000000000000000000> }LINE 11
B<000000000000000000000000> }LINE 12
B<000000000000000000000000> }LINE 13
B<000000000000000000000000> }LINE 14
B<000000000000000000000000> }LINE 15

```

```

990
991
992
993
994
995 012752* 013106*
996 012754* 177748*
997 012756* 013148*
998 012760* 177748*
999 012762* 000000*
1000 012764* 177703*
1001 012766* 000000*
1002 012770* 000000*
1003 012772* 006158*
1004 012774* 002158*
1005 012776* 014013*
1006
1007 013000* 000004*
1008 013002* 000000*
1009 013004* 000000*
1010 013006* 000000*
1011 013010* 000000*
1012

```

```

; THIS IS A SAMPLE OF WHAT IS LOADED INTO
; THE DV11 SECONDARY REGISTER FOR THE LINES
; THAT WERE SELECTED TO RUN.
; HOWEVER THE BUS ADDRESS AND CHAR CNTS MAY CHANGE.
FUNC.X: TXBAP 000 TX PRI BUS ADDR.
-40 003 TX PRI BYTE CNT
TXBAS -40 003 TX ALT BUS ADDR.
-40 003 TX ALT BYTE CNT
RXB 004 RX BUS ADDR.
-75 002 TX BYTE CNT
0 006 TX BCC REG
0 007 RX BCC REG
TX-CONTROL-TABLE 010 TX CNTRL TABLE
RX-CONTROL-TABLE 011 RX CNTRL TABLE
DLE*HIBYTE+CRCL6+STRIP.SYNC+IDLE MARK 012 TX DLE REGISTER STRIP LEADING SYNC; IDLE MARK
TX-GO 013 LINE STATE REGISTER
MODE0 014 TX MODE BIT REG
MODE0 015 RX MODE BIT REG
0 016 LINE PROTOCOL REG
0 017 RX CONTROL BYTE STORAGE

```

```

; THIS IS AN EXACT IMAGE
; OF WHAT DATA SHOULD BE FOUND IN ANY COMPLETED
; DV11 RECEIVER BUFFER. (NOTE: DLE=30(8))

```

```

013012* 000 001 002
013014* 010 011 012
013021* 020 021 022
013030* 030 031 032
013033* 040 041 042
013063* 050 051 052
013073* 060 061 062
013077* 070 071 072

```

```

DATA.TABLE:
-BYTE 0,1,2,3,4,5,6
-BYTE 10,11,12,13,14,15,16
-BYTE 20,21,22,23,24,25,26
-BYTE DLE,30,DLE,31,DLE,32,DLE,33,DLE,34,DLE,35,36,DLE,37
-BYTE 40,41,42,43,44,45,46,47
-BYTE 50,51,52,53,54,55,56,57
-BYTE 60,61,62,63,64,65,66,67
-BYTE 70,71,72,73,74,75,76;-

```

```

; THIS THE DATA THAT THE TRANSMITTERS SEND.

```

```

(1) 013106* 000 001 002
(2) 013106* 010 011 012
(3) 013116* 020 021 022
(4) 013126* 030 031 032
(5) 013146* 040 041 042
(6) 013146* 050 051 052
(7) 013156* 060 061 062
(8) 013176* 070 071 072

```

```

TXBAP:
-BYTE 0,1,2,3,4,5,6,7
-BYTE 10,11,12,13,14,15,16,17
-BYTE 20,21,22,23,24,25,26,27
-BYTE 30,31,32,33,34,35,36,37
TXBAS:
-BYTE 40,41,42,43,44,45,46,47
-BYTE 50,51,52,53,54,55,56,57
-BYTE 60,61,62,63,64,65,66,67
-BYTE 70,71,72,73,74,75,76,77

```

.EVEN

```
1013 ;RECEIVER BUFFERS FOR DEVICES #3 AND #4.  
1014 RXB30: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #0  
1015 RXB31: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #1  
1016 RXB32: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #2  
1017 RXB33: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #3  
1018  
1019 MEXIT  
1020 RXB34: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #4  
1021 RXB35: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #5  
1022 RXB36: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #6  
1023 RXB37: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #7  
1024 MEXIT  
1025 RXB310: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #10  
1026 RXB311: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #11  
1027 RXB312: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #12  
1028 RXB313: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #13  
1029 MEXIT  
1030 RXB314: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #14  
1031 RXB315: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #15  
1032 RXB316: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #16  
1033 RXB317: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #3 LINE #17  
1034 MEXIT  
1035 RXB40: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #0  
1036 RXB41: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #1  
1037 RXB42: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #2  
1038 RXB43: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #3  
1039 MEXIT  
1040 RXB44: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #4  
1041 RXB45: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #5  
1042 RXB46: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #6  
1043 RXB47: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #7  
1044 MEXIT  
1045 RXB410: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #10  
1046 RXB411: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #11  
1047 RXB412: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #12  
1048 RXB413: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #13  
1049 MEXIT  
1050 RXB414: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #14  
1051 RXB415: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #15  
1052 RXB416: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #16  
1053 RXB417: .BLKB 74 ;THIS IS RECEIVER BUFFER FOR DEVICE #4 LINE #17  
1054 MEXIT
```

```
1055 017006*  
1056  
1057  
1058  
1059 017006* 004567 173436  
1060 017012* 000000 006206*  
1061 017012* 000000 173214  
1062 017012* 000000  
1063 017024* 000224*  
1064  
1065 017026* 004567 173416  
1066 017032* 000000 010226*  
1067 017036* 004567 173174  
1068 017042* 000000  
1069 017044* 000236*  
1070  
1071 017046* 004567 173376  
1072 017052* 000000 013206*  
1073 017056* 004567 173154  
1074 017062* 000000  
1075 017064* 000250*  
1076  
1077 017066* 004567 173356  
1078 017072* 000000 015106*  
1079 017076* 004567 173134  
1080 017104* 000000  
1081 017104* 000262*  
1082  
1083  
1084 017106*  
1085 017106* 006206*  
1086 017110* 006302*  
1087 017112* 006376*  
1088 017114* 006472*  
1089 017116* 006576*  
1090 017120* 006672*  
1091 017122* 006766*  
1092 017124* 007066*  
1093 017126* 007206*  
1094 017130* 007302*  
1095 017132* 007376*  
1096 017134* 007472*  
1097 017136* 007616*  
1098 017140* 007712*  
1099 017142* 010006*  
1100 017144* 010102*  
1101  
1102 017146* 010226*  
1103 017150* 010322*  
1104 017152* 010416*  
1105 017154* 010512*  
1106 017156* 010636*  
1107 017160* 010732*  
1108 017162* 011026*  
1109 017164* 011122*  
1110 017166* 011246*  
  
LNKTAB:  
; ALL INTERRUPTS WILL BE CHANNLED THROUGH  
; THE APPROPRIATE CALL.  
;  
JSR R5,RXISR ;GOTO RECEIVER ISR FOR DEVICE #1.  
;WORD SDVSCR,RXB10 ;GOTO TRANSMITTER ISR FOR DEVICE #1  
R5,TXISR  
;WORD DVSCR  
XXX1: ;WORD LINES1 ;END PASS FLAG.  
;  
JSR R5,RXISR ;GOTO RECEIVER ISR FOR DEVICE #2.  
;WORD SDVSCR,RXB20 ;GOTO TRANSMITTER ISR FOR DEVICE #2  
R5,TXISR  
;WORD DVSCR  
XXX2: ;WORD LINES2 ;END PASS FLAG.  
;  
JSR R5,RXISR ;GOTO RECEIVER ISR FOR DEVICE #3.  
;WORD SDVSCR,RXB30 ;GOTO TRANSMITTER ISR FOR DEVICE #3  
R5,TXISR  
;WORD DVSCR  
XXX3: ;WORD LINES3 ;END PASS FLAG.  
;  
JSR R5,RXISR ;GOTO RECEIVER ISR FOR DEVICE #4.  
;WORD SDVSCR,RXB40 ;GOTO TRANSMITTER ISR FOR DEVICE #4  
R5,TXISR  
;WORD DVSCR  
XXX4: ;WORD LINES4 ;END PASS FLAG.  
;  
MEXIT  
BUFFER TABLE:  
RXB10 ;BUFFER FOR DEVICE #1 LINE #0  
RXB11 ;BUFFER FOR DEVICE #1 LINE #1  
RXB12 ;BUFFER FOR DEVICE #1 LINE #2  
RXB13 ;BUFFER FOR DEVICE #1 LINE #3  
RXB14 ;BUFFER FOR DEVICE #1 LINE #4  
RXB15 ;BUFFER FOR DEVICE #1 LINE #5  
RXB16 ;BUFFER FOR DEVICE #1 LINE #6  
RXB17 ;BUFFER FOR DEVICE #1 LINE #7  
RXB110 ;BUFFER FOR DEVICE #1 LINE #10  
RXB111 ;BUFFER FOR DEVICE #1 LINE #11  
RXB112 ;BUFFER FOR DEVICE #1 LINE #12  
RXB113 ;BUFFER FOR DEVICE #1 LINE #13  
RXB114 ;BUFFER FOR DEVICE #1 LINE #14  
RXB115 ;BUFFER FOR DEVICE #1 LINE #15  
RXB116 ;BUFFER FOR DEVICE #1 LINE #16  
RXB117 ;BUFFER FOR DEVICE #1 LINE #17  
  
RXB20 ;BUFFER FOR DEVICE #2 LINE #0  
RXB21 ;BUFFER FOR DEVICE #2 LINE #1  
RXB22 ;BUFFER FOR DEVICE #2 LINE #2  
RXB23 ;BUFFER FOR DEVICE #2 LINE #3  
RXB24 ;BUFFER FOR DEVICE #2 LINE #4  
RXB25 ;BUFFER FOR DEVICE #2 LINE #5  
RXB26 ;BUFFER FOR DEVICE #2 LINE #6  
RXB27 ;BUFFER FOR DEVICE #2 LINE #7  
RXB210 ;BUFFER FOR DEVICE #2 LINE #10
```


DVARO.P11 12-OCT-78 11:57
 DATERS= 104404 273# 597
 DEBUC = ***** U 1059
 DLE = 000030 1005 1012#
 DROP = 000356R 340
 DSCARD= 000010 316# 343
 DVIDI = 000014R 230# 334
 DVLCHR = 000004 301# 421*
 DVNSR = 000014 306# 886
 DVRES = 000016 307#
 DVRIC = 000002 300# 924
 DVSF = 000012 305#
 DVSR = 000010 304#
 DVSRSH= 000007 302# 397* 424* 455* 463 466 469 473* 475* 476* 501* 901* 907*
 303# 398* 419* 428* 900* 938*
 304# 420* 425* 426 429* 454* 459* 462* 465* 468* 474* 904* 939*
 305#
 306# 471 489 500
 307# 583
 308# 608
 309# 972#
 310# 546*
 311#
 312# 471 995#
 313# 583
 314# 337 608
 315# 972#
 316# 546*
 317#
 318# 995#
 319# 488 497
 320#
 321# 1005
 322# 375* 391* 458
 323# 531 548
 324# 245#
 325# 611#
 326# 711#
 327# 241#
 328# 217# 1005
 329# 270#
 330# 371 551 562 1063
 331# 346 1069
 332# 347 1079
 333# 348 1075
 334# 349 557 1081
 335# 1055#
 336# 627 628 629 630 631 632 633 635 636 637 638 639
 337# 641 642 643 644 645 646 647 648 649 650 651 652
 338# 735 736 737 738 739 740 742 743 744 745 746 747 748
 339# 749 750 751 752 753 754 755 756 757 758 760 761 762
 340# 768 769 770 771 772 773 774 775 776 777 778 779 780
 341# 311# 662 666 667 668 669 670 671 672 772 781 782 783
 342# 784 785 786 787 788 789 681 682 788 796 797 798
 343# 799 800 801 802 678 679 680 681 682 788 796 797 798
 344# 313# 683 687 688 689 690 691 692 693 803 813 814 815
 345# 816 817 818 819 700 701 702 703 704 820 829 830 831
 346# 694 698 699 700 701 702 703 704 820 829 830 831

MODE1 = 000040
 MODE2 = 000100
 MODE3 = 000140
 MODE4 = 000200
 MODE5 = 000240

DVARO.P11 12-OCT-78 11:57
 DLE = 000030 1005 1012#
 DROP = 000356R 340
 DSCARD= 000010 316# 343
 DVIDI = 000014R 230# 334
 DVLCHR = 000004 301# 421*
 DVNSR = 000014 306# 886
 DVRES = 000016 307#
 DVRIC = 000002 300# 924
 DVSF = 000012 305#
 DVSR = 000010 304#
 DVSRSH= 000007 302# 397* 424* 455* 463 466 469 473* 475* 476* 501* 901* 907*
 303# 398* 419* 428* 900* 938*
 304# 420* 425* 426 429* 454* 459* 462* 465* 468* 474* 904* 939*
 305#
 306# 471 489 500
 307# 583
 308# 608
 309# 972#
 310# 546*
 311#
 312# 471 995#
 313# 583
 314# 337 608
 315# 972#
 316# 546*
 317#
 318# 995#
 319# 488 497
 320#
 321# 1005
 322# 375* 391* 458
 323# 531 548
 324# 245#
 325# 611#
 326# 711#
 327# 241#
 328# 217# 1005
 329# 270#
 330# 371 551 562 1063
 331# 346 1069
 332# 347 1079
 333# 348 1075
 334# 349 557 1081
 335# 1055#
 336# 627 628 629 630 631 632 633 635 636 637 638 639
 337# 641 642 643 644 645 646 647 648 649 650 651 652
 338# 735 736 737 738 739 740 742 743 744 745 746 747 748
 339# 749 750 751 752 753 754 755 756 757 758 760 761 762
 340# 768 769 770 771 772 773 774 775 776 777 778 779 780
 341# 311# 662 666 667 668 669 670 671 672 772 781 782 783
 342# 784 785 786 787 788 789 681 682 788 796 797 798
 343# 799 800 801 802 678 679 680 681 682 788 796 797 798
 344# 313# 683 687 688 689 690 691 692 693 803 813 814 815
 345# 816 817 818 819 700 701 702 703 704 820 829 830 831
 346# 694 698 699 700 701 702 703 704 820 829 830 831

MODE6 = 000300
 MODE7 = 000340

MODNAM = 000000R 224#
 MODSP = 000224R 238# 271#
 MRESCT = 004000 217# 393# 909 958
 MSGNS = 104403 273# 607
 MSGS = 104402 273#
 MSGS = 104401 273#
 M.WODE = 010000 217#
 NPR.IE = 020000 273#
 NULL = 000000 273# 548
 OPEN = 000000 273# 233 234 251 252 253 254 255 256 258
 275# 262 264 265 267 268 269 273#
 DTOAS = 104420 273#
 DVFLO = 000200 217# 878
 PA = 000300R 320# 472 490 501
 PASCNT = 000034R 273#
 PING.P = 000274R 318# 332* 364 372 379 402 563 582*
 PIRGS = 000004 273#
 POPSP = 005726 273#
 POPSP2 = 022626 273#
 PRI.EA = 012704R 471# 906 967#
 PRI.PA = 012706R 472# 907 969#
 PRTY = 000000 273#
 PRTY0 = 000000 273#
 PRTY1 = 000040 273#
 PRTY2 = 000100 273#
 PRTY3 = 000140 273#
 PRTY4 = 000200 273#
 PRTY5 = 000240 229 273#
 PRTY6 = 000300 273#
 PRTY7 = 000340 273#
 PS = 177776 273#
 PSM = 177776 273#
 PUSH2 = 005746 273#
 RANDS = 104417 273#
 RANNUM = 000054R 247#
 RENABL = 020000 217# 421
 RESRRT = 000340R 265# 333# 550 587 941
 RESVNC = 000002 217#
 RES1 = 000056R 249#
 RES2 = 000060R 250#
 RSRT = 000110R 266#
 RXB = 000000 996# 1012#
 RXB10 = 006206R 758# 1060
 RXB11 = 006302R 759# 1086
 RXB110 = 007066R 789# 1093
 RXB111 = 007302R 790# 1094
 RXB112 = 007376R 791# 1095
 RXB113 = 007472R 792# 1096
 RXB114 = 007712R 804# 1097
 RXB115 = 007712R 805# 1098

