

.REM

IDENTIFICATION  
-----

PRODUCT CODE: AC-E905B-MC  
PRODUCT NAME: CXVTAB0 VT20 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

VTA IS AN IOMOD THAT EXERCISES UP TO FOUR VT20'S (8 DL11 LINES). TEST IS INTENDED TO BE A DATA HANDLING ROUTINE USED IN CONNECTION WITH TEST 21 OF WAINDEC-11-DBVTA (PREVIOUSLY LOADED AND RUNNING IN THE VT20'S PDPI1-05). DATA IS ENTERED AT EACH RECEIVED TUBE AND SET INTO THE CONTINUOUS TRANSMIT MODE. THIS DATA IS THEN RECEIVED AND RETRANSMITTED BY THE VT20 HOST COMPUTER (THIS CONTIGUOUS ADDRESSES ALL LINES SELECTED FOR TEST (UP TO 8 DL11'S WITH CONTIGUOUS ADDRESSES AND VECTORS) CAN BE ACTIVATED AND RUN CONCURRENTLY. ALL TRANSMIT AND RECEIVE ERRORS ARE REPORTED AND ON THE CONSOLE ITY. NO DATA ERRORS ARE REPORTED BY THIS MODULE.

2.

REQUIREMENTS

HARDWARE: AT LEAST ONE VT20

STORAGE: VTA REQUIRES:

- 1. DECIMAL WORDS: 2296
- 2. OCTAL WORDS: 04370
- 3. OCTAL BYTES: 10760

3.

PASS DEFINITION

ONE PASS OF THE VTA MODULE CONSISTS OF CONTINUOUSLY RECEIVING AND TRANSMITTING THE DATA ENTERED ON ALL SELECTED LINES FOR THE PERIOD DEFINED BELOW.

4.

EXECUTION TIME

EXECUTION TIME VARIES WITH THE NUMBER OF JOBS (MODULES) ACTIVATED. THE BAUD RATE AND THE NUMBER OF TUBES BEING EXERCISED. HOWEVER, THIS MODULE RUNNING ALONE WILL TAKE ABOUT 20 SECONDS FOR ONE TUBE AND UP TO 60 SECONDS FOR 8 TUBES. BAUD RATES LESS THAN 2400 REQUIRE PROGRAM MODIFICATION WHICH INCREASES EXECUTION TIME. SEE SECTION 6.B.

5.

CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA:175610, VCT:340, BR1:5, BR2:0, DVC:1

REQUIRED PARAMETERS:

VCT:VECTOR ADDRESS OF FIRST DL11 IF NOT 340  
DVC:NO OF DL11'S (TUBES) IF GREATER THAN 1

6. DEVICE SETUP

A. THE USER MUST LOAD AND START MODULE TEST 21 OF MAINDEC-11-DBVTA IN THE VT20 PDP11/05 IN ORDER FOR THIS MODULE TO EXERCISE. CONSULT THE ABOVE DOCUMENT AND COMPLY WITH THE OPERATING INSTRUCTIONS FOR TEST 21 (SECTION 26). THIS DEC/X11 MODULE EXPECTS THE USER TO ENTER DATA ON EACH TUBE AND SET EACH TUBE IN THE CONTINUOUS TRANSMIT MODE. THIS SETTING IS TAKEN AFTER THE DEC/X11 EXERCISER HAS BEEN STARTED BY THE SLIP, RUN, IS, COMMAND. A TYPICAL USER ACTION ON EACH SELECTED TUBE WILL BE AS FOLLOWS:

KEY	FUNCTION
CTRL E	CLEAR SCREEN
CTRL W	GENERATE WORST CASE CHARACTER PATTERN ON TOP OF SCREEN
CTRL T	CONTINUOUS TRANSMIT TO DEC/X11 MODULE (DEC/X11 MODULE WILL RECEIVE DATA AND TRANSMIT IT BACK TO BOTTOM OF SCREEN)

NOTE: IF THE CHARACTER PATTERN FAILS TO RETURN ON THE BOTTOM OF THE SCREEN AFTER ONE 'CTRL T', THEN RETRY AFTER END PASS. IS REPORTED FOR THIS DEC/X11 MODULE (DATA IS STILL ARE TURNED OFF SECONDS BEFORE END PASS. MSG). IF DATA IS STILL NOT RETURNED FROM HOST COMPUTER (DEC/X11 SYSTEM) THEN VERIFY THE VT20 HOST COMPUTER BY RUNNING MAINDEC-11-DZVTE.

B. IF BAUDS RATES LOWER THAN 2400 ARE USED THEN ONE OF THE VALUES BELOW MUST BE PLUGGED IN LOCATION 'COUNT1'. (VTA 1660) TO AVOID POSSIBLE 'DL11 HUNG' ERRORS. NOTE THAT EXECUTION TIME INCREASES BY ABOUT 30 SECONDS FOR EVERY 1 COUNT.

BAUD RATE	VALUE
1200	2
900	2
300	2
150	3
110	4

USE THE "MOD" COMMAND TO ALTER LOCATION 'COUNT1' (VTA 1660)

7. MODULE OPERATION

7.1 TEST SEQUENCE

A. START: USING THE DEVICE SELECTION PARAMETER "DVID1" THIS SECTION OF CODE SETS UP THE VECTORS OF ALL SELECTED LINES TO POINT TO THE APPROPRIATE JSR IN THE JSR LINKING TABLE.

P. SETCSP: THIS PIECE OF CODE INSERTS THE PROPER CSR ADDRESS OF EACH ACTIVE LINE INTO THE THIRD

WORD OF EACH JSR TABLE ENTRY.

C. SETUP: THIS CODE INITIALIZES ALL TABLES, BUFFERS, FLAGS, AND COUNTERS.

D. STRTUP: THIS CODE TURNS ON THE INTERRUPT ENABLER FOR EACH SELECTED RECEIVER.

E. TIMR: THIS CODE IS AN 'END PASS' TIMER LOOP VIA 'BREAKS' TO THE MONITOR. THE PROGRAM LEAVES THIS LOOP TO RESTART LINES ON TRANSMIT AFTER COMPLETION OF RECEIVING A BLOCK OF DATA AND TO PREPARE FOR THE 'END PASS' MESSAGE AND TO REPORT RECEIVER ERRORS IF ANY.

F. XSTRT: RESTARTS EACH LINE TRANSMITTING THAT HAS RECEIVED A BLOCK OF DATA (CHARACTER '014' TERMINATES A BLOCK OF DATA).

G. TMOUT: THIS CODE IS ENTERED WHEN THE 'END PASS' MESSAGE IS CALLED FOR. IT PROVIDES TIME VIA 'BREAKS' FOR ALL LINES TO BECOME IDLE AT COMPLETION OF RECEIVE. REPORTS RECEIVER ERRORS IF ANY AND CHECKS THAT AT LEAST ONE GOOD LINE IS LEFT TO BE RESTARTED.

H. HNGTST: AFTER WAITING ENOUGH TIME FOR ALL LINES TO BECOME IDLE ALL LINES ARE EXPECTED TO HAVE COMPLETED RECEIVING IF NOT THE LINE IS REPORTED HUNG AND DROPPED (SEE SECTION 6 FOR RATIO OF LINES LESS THAN 2400). 'END PASS' MESSAGE IS NOW TYPED.

I. RESTR: THIS CODE INITIALIZES QUEUES AND TIMERS AND STARTS UP ANY LINE TO RECEIVE PREVIOUSLY REPORTED AS HUNG THEN GOES AND STARTS UP ALL OTHER LINES TRANSMITTING.

J. RINT: THE RECEIVER SERVICE ROUTINE STORES DATA, AND CHECKS FOR RECEIVER ERRORS WHICH ARE STORED FOR BACKGROUND REPORTING. IT ALSO LOOKS FOR THE TERMINATING CHARACTER OF '014'. WHEN RECEIVED, IT SETS UP FOR TRANSMIT.

K. TINT: THE TRANSMITTER SERVICE ROUTINE SIMPLY QUEUES UP THE REQUEST FOR SERVICE IN A FIFO QUEUE UPDATES THE QUEUE POINTER WITH RETURN CONTROL BACK TO THE MONITOR WITH A 'PIRQ'. THE ELEMENT THAT GETS STORED IN THE QUEUE IS A POINTER TO THE INTERRUPTING CSR ADDRESS. THE ACTUAL SERVICE IS DONE LATER WHERE THE SERVICE CODE IS EXECUTED AT LEVEL 0.

L. TSERV: THIS CODE RETRIEVES A POINTER FROM THE FIFO QUEUE AND BUILDS THE CSR ADDRESS. STATUS IS CHECKED AND ERRORS REPORTING IF THE NEXT CHARACTER IS THE TERMINATING CODE (014) THE LINE WILL THEN BE INITIALIZED

TO RECEIVE. IN ANY EVENT THE NEXT CHARACTER WILL BE OUTPUTTED AND AN EXIT BACK TO THE MONITOR PERFORMED.

M. RERCK: THIS CODE REPORTS ANY RECEIVER ERRORS WHICH HAD BEEN DUMPED INTO THE ERROR QUEUE 'EQ'.

7.2 DESCRIPTION OF TABLES, QUEUES, AND BUFFERS

A. DLSTUS: 8 WORD TABLE WHICH SPECIFIES WHAT EACH LINE IS DOING: NOT SELECTED (0), TRANSMITTING (1), RECEIVING (2), SELECTED AND WAITING FOR USER ACTION ON VT20 KEYBOARD (3), WAITING FOR TRANSMIT AFTER RECEIVE (-1). ALL LINES SHOULD REACH THIS STATE BEFORE 'END PASS'.

F. RCVSW: 8 WORD TABLE INDICATING THAT A LINE HAS FOUND THE SYNC OR START CODE OF 377 WHICH BEGINS A BLOCK OF DATA: 0 = NO SYNC, 1 = SYNC HAS BEEN MADE.

C. ERTAB: 8 WORD TABLE WHICH INDICATES THE NUMBER OF ERRORS THAT HAVE OCCURRED DURING EACH TRANSFER.

D. TQ: 8 WORD FIFO QUEUE FOR TRANSMITTER SERVICE. LOADED WITH A POINTER TO THE CSR ADDRESS AND UNLOADED DURING DEFERRED XMTR SERVICE.

E. EQ: 48 WORD FIFO QUEUE FOR RECEIVER ERROR REPORTING. LOADED WITH THE BAD LINE'S CSR ADDRESS AND STATUS AND DATA - 2 WORDS PER ERROR.

F. XRB0-7: 8-390 BYTE RECEIVE/TRANSMIT DATA BUFFERS

G. JSRTAB: 64 WORD TABLE THAT CONTAINS 16 JSR INSTRUCTIONS WITH TWO TRAILING ARGUMENTS. EACH RECEIVER AND EACH XMTR HAS AN ASSIGNED JSR IN THE TABLE OF THE FOLLOWING FORMAT:

```
JSR R5,RINT(TINT)
0
N
```

WHERE THE 0 GETS OVERLAPPED WITH THE ADDRESS OF THE CSR FOR LINE N AND N IS THE LINE NO. IN OCTAL TIMES TWO (00-16)

8. OPERATOR OPTIONS  
-----

A. THE USER CAN MODIFY (VIA I4) "DVIDI" TO SELECT OR DESELECT INDIVIDUAL DL11-S.

B. THE USER CAN USE THE "MOD" COMMAND TO DUMP THE TABLES  
OR BUFFERS DESCRIBED IN 7.2 TO OBTAIN MORE DETAILED  
ERROR INFORMATION.

9. ERROR PRINTOUTS  
-----

9.1 ERROR FORMAT - RECEIVE

CSRA = CSR ADDRESS  
CSRC = DBR WORD AS FOLLOWS:

- BIT 15 = DL11 ERROR
- \*\*BIT 14 = OVERRUN
- BIT 13 = FRAMING
- BIT 12 = PARITY
- BIT 7-0 = DATA RECEIVED

OCTAL WORD FOLLOWING ERROR DEFINED AS FOLLOWS:

- BIT 3 = ILLEGAL INTERRUPT - INT EN OR DONE NOT SET
- BIT 2 = ILLEGAL SYNC CHARACTER - COULD NOT SYNC ON RECEIVE
- (1ST NON-ZERO CHAR WAS NOT THE #377 CODE)
- BIT 1 = DL11 DROPPED FROM MODULE - 3 ERRORS OCCURRED ON
- ANY DATA BLOCK
- BIT 0 = DL11 IS HUNG - DL FAILED TO RECEIVE A BLOCK OF
- DATA IN WORST CASE TIME.

\*\*NOTE:

OVERRUN ERRORS WILL START OCCURRING ON LARGE SYSTEMS  
WHERE BUS ACTIVITY IS HIGH AS THE NUMBER OF V120 TUBES  
ACTIVATED IS INCREASED. I.E. IF 8 TUBES AT 9600 BAUD  
ARE SELECTED THE WORDS RECEIVED IN INTERRUPT SERVICE  
TIME IS APPROXIMATELY 140 MICRO-SECONDS. THIS MEANS THAT  
IF ALL OTHER I/O BUS ACTIVITIES PLUS SOFTWARE SLOP INHIBITS  
A RECEIVE INTERRUPT FROM BEING SERVICED IN 140 MICRO-SECONDS  
THEN RECEIVED OVERRUN ERRORS WILL START OCCURRING AT THE DL11 LINE  
ELECTRICALLY MOST DISTANT FROM THE PROCESSOR.

VTAB DEC/X11 SYSTEM EXERCISER MODULE MACY11 30A(1052) 12-OCT-78 17:09 PAGE 8  
XVTAB0.P11 12-OCT-78 12:24

SEQ 0007

9.2 ERROR FORMAT - TRANSMIT

CSRA = CSR ADDRESS  
CSRC = CSR CONTENTS AS FOLLOWS:

BIT 7 = XMITR READY  
BIT 6 = XMITR INTERRUPT ENABLED

OCTAL WORD FOLLOWING ERROR DEFINED AS FOLLOWS:

BIT 1 = XMITR IS HUNG - DL FAILED TO INTERRUPT  
BIT 0 = DLI1 DROPPED FROM MODULE - OCCURES ON ABOVE ERROR

```

307 000000* IOMOD <VTAB> 175610,340,5,4,2,64
308 000000* MODULE 140000,VTAB,175610,340,2,64
309 000000* .TITLE VTAB DEC/X11 SYSTEM EXERCISER MODULE
310 000000* .DPXCOM VERSION 6 23-MAY-78
311 000000* .LIST
312 *****
313 BEGIN:
314 MODNAM: 052126 041101 040 .ASCII /VTAB / ;MODULE NAME.
315 XFLAG: 0 .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
316 ADDR: 175610+0 .WORD ;1ST DEVICE ADDR.
317 VECTOR: 340+0 .WORD ;1ST DEVICE VECTOR.
318 BR1: .BYTE PRTV5+0 ;1ST BR LEVEL.
319 BR2: .BYTE PRTV+0 ;2ND BR LEVEL.
320 DVID1: +1 .WORD ;DEVICE INDICATOR 1.
321 SR1: OPEN .WORD ;SWITCH REGISTER 1.
322 SR2: OPEN .WORD ;SWITCH REGISTER 2.
323 SR3: OPEN .WORD ;SWITCH REGISTER 3.
324 SR4: OPEN .WORD ;SWITCH REGISTER 4.
325 *****
326 STAT: 140000 .WORD ;STATUS WORD.
327 INIT: START .WORD ;MODULE START ADDR.
328 SPCHT: MODSP .WORD ;MODULE STACK POINTER.
329 PASCT: 0 .WORD ;PASS COUNTER.
330 ICOUNT: 0 .WORD ;NO. ITERATIONS PER PASS=2
331 SDFCNT: 0 .WORD ;LOC TO COUNT ITERATIONS
332 HRDCNT: 0 .WORD ;LOC TO SAVE TOTAL SOFT ERRORS
333 SDFPAS: 0 .WORD ;LOC TO SAVE TOTAL HARD ERRORS
334 HRDPAS: 0 .WORD ;LOC TO SAVE SOFT ERRORS PER PASS
335 SYSCNT: 0 .WORD ;LOC TO SAVE HARD ERRORS PER PASS
336 RANMUH: 0 .WORD ;NO. OF SVS ERRORS ACCUMULATED
337 CONFIG: 0 .WORD ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
338 RES: 0 .WORD ;RESERVED FOR MONITOR USE
339 SVRO: OPEN .WORD ;LOC TO SAVE R0.
340 SVR1: OPEN .WORD ;LOC TO SAVE R1.
341 SVR2: OPEN .WORD ;LOC TO SAVE R2.
342 SVR3: OPEN .WORD ;LOC TO SAVE R3.
343 SVR4: OPEN .WORD ;LOC TO SAVE R4.
344 SVR5: OPEN .WORD ;LOC TO SAVE R5.
345 SVR6: OPEN .WORD ;LOC TO SAVE R6.
346 CSRA: OPEN .WORD ;ADDR OF CURRENT CSR.
347 SBADR: 0 .WORD ;ADDR OF GOOD DATA, OR
348 WASADR: OPEN .WORD ;ADDR OF BAD DATA, OR
349 ASADR: OPEN .WORD ;ADDR OF BAD DATA, OR
350 ERRTYP: OPEN .WORD ;STATUS REG CONTENTS.
351 ASB: OPEN .WORD ;EXPECTED DATA.
352 ANAS: OPEN .WORD ;ACTUAL DATA.
353 WDT0: OPEN .WORD ;RESTART ADDRESS AFTER END OF PASS
354 WDFR: OPEN .WORD ;WORDS FROM MEMORY PER ITERATION
355 WDFR: OPEN .WORD ;NO. OF INTERRUPTS PER ITERATION
356 IDNUM: 64 .WORD ;MODULE IDENTIFICATION NUMBER=64
357 .REPT SPSIZ ;MODULE STACK STARTS HERE.
358 .MLIST

```

```

363 .WORD 0
364 .LIST
365 .ENBR
366 000224*
367 *****
368 ;THIS ROUTINE SETS UP THE VECTORS FOR ALL SELECTED LINES TO POINT
369 ;TO THE APPROPRIATE JSR IN THE JSR LINK TABLE.
370 *****
371 START: MOV #390,INTR ;390 INTERRUPTS/ITERATION
372 MOV #390,WDT0 ;390 WORDS TO NEXT ITERATION
373 MOV VECTOR,R0 ;SET R0 TO POINT TO THE 1ST VECTOR
374 MOV DVID1,R1 ;LOAD R1 WITH DEVICE SELECTION PARAMETER
375 MOV #JSRTAB,R2 ;SETUP R2 TO POINT TO JSR TABLE
376 ASR R1 ;SHIFT SELECT BIT INTO "C"
377 BCC ;BR IF NOT SELECTED
378 ADD #390,WDT0 ;390 MORE
379 ADD #390,INTR ;390 MORE INTS.
380 MOV R2,(R0)+ ;SET UP RCVR INTR POINTER
381 MOV B1,(R0)+ ;SET UP RCVR PRIORITTY LEVEL
382 INC R0 ;MOVE POINTER
383 MOV #R2,(R0)+ ;POINT R2 TO XMTN ENTRY IN JSRTAB
384 MOV B1,(R0)+ ;SET UP INTR PRIORITY LEVEL
385 INC R0 ;MOVE POINTER
386 ADD #390,R2 ;POINT R2 TO RCVR ENTRY FOR NEXT LINE
387 CMP #SETQTS,R2 ;IS THE POINTER AT THE END OF THE TABLE?
388 BNE ;BR IF NOT
389 SETCSR ;GO SET UP CSR ADDRESSES
390 ADD #10,R0 ;ADVANCE VECTOR POINTER
391 ADD #20,R2 ;ADVANCE JSR TABLE POINTER
392 BR 2$ ;GO CHECK FOR END OF TABLE
393 2$:
394 ;THIS ROUTINE SETS UP THE JSR TABLE SUCH THAT THE APPROPRIATE
395 ;CSR ADDRESS IS INCLUDED AS THE 3RD WORD OF EACH ENTRY
396 *****
397 SETCSR: MOV ADDR,R0 ;GET THE FIRST CSR ADDRESS INTO R0
398 MOV DVID1,R1 ;LOAD R1 WITH THE DEVICE SELECTION PARAMETER
399 BNE ;BR IF SOMETHING SELECTED
400 ENDS,BEGIN ;DROP MODULE-NOHING SELECTED
401 MOV #JSRTAB+4,R2 ;POINT R2 TO CSR ADDRESS ENTRY
402 ASR R1 ;SHIFT SELECT BIT INTO "C"
403 BCC ;BR IF LINE NOT SELECTED
404 MOV R0,(R2) ;PUT RCVR CSR ADDRESS IN TABLE
405 ADD #4,R0 ;GENERATE XMTN CSR ADRS IN R0
406 MOV R0,(R2) ;PUT XMTN CSR ADDRESS IN JSR TABLE
407 ADD #4,R0 ;GENERATE RCVR CSR ADRS IN R0
408 MOV R0,(R2) ;POINT TO RCVR SLOT IN JSR TABLE
409 CMP #SETQTS+4,R2 ;IS POINTER BEYOND END OF TABLE?
410 BNE ;BR IF NOT
411 SETUP ;GO SETUP FOR RECEIVE
412 ADD #10,R0 ;UPDATE CSR ADDRESS
413 ADD #20,R2 ;UPDATE JSR TABLE POINTER
414 BR 3$ ;GO TEST FOR END OF TABLE
415 3$:
416 *****
417 ;THIS CODE CLEARS BUFFERS AND TABLES AND INITIALIZES FLAGS
418 *****

```



```

419 000442* C05000
420 000444* 016060
421 000452* 062760 002040* 002014* SETUP: CLR R0
422 000460* 062700 000003 002014* 1$: ADD #2,RBPO(R0) ;SET UP BUFFER POINTER-RCVR
423 000464* 022700 000002 ;ADVANCE POINTER OVER NULL CHARS
424 000470* 013655 ;ADVANCE TO NEXT BUFFER
425 000472* 013655 ;CLEAR THIS LOCATION
426 000476* 065020 002320* 2$: CLR #20,R0 ;HAVE WE SET UP 8 BUFFERS
427 000500* 022700 010400* ;BR TP MORE
428 000504* 001374 ;SETUP 1ST RCVR SOFTWARE SWITCH ADRS
429 000506* 012701 002100* ;SETUP THIS LOCATION
430 000512* 012700 002120* ;HAVE WE CLEARED ALL BUFFERS?
431 000516* 005020 ;BR IF NOT
432 000520* 005021 ;SETUP 1ST ERROR COUNTERS ADRS
433 000522* 020027 002140* 3$: MOV #ERTAB,R0 ;SETUP 1ST ERROR COUNTERS ADRS
434 000526* 001373 ;CLR THIS COUNTER
435 000530* 004767 010044 ;CLR THIS RCVR SOFTWARE SWITCH ADRS
436 000534* 012700 177776 ;HAVE ALL ERROR COUNTERS BEEN CLEARED
437 000540* 005001 ;BR IF NOT
438 000542* 016702 177246 ;GO SET UP Q'S & TIMERS
439 000546* 062700 000002 ;INITIALLY SET TO MINUS TWO
440 000552* 022700 000020 4$: ADD #2,R0 ;CLEAR INITIAL DEVICE COUNT
441 000556* 010167 001210 ;GET DEVICE SELECTION PARAMETER
442 000560* 001404 ;ADVANCE TABLE OFFSET
443 000564* 000412 ;HAVE LOOKED AT 8 DL'S?
444 000566* 006202 002060* 5$: BR #ACTDEV ;SAVE THE COUNT OF ACTIVE LINES
445 000570* 103403 ;ACTIVE TABLE SET UP
446 000574* 000103 002060* 6$: BCS #0,R1 ;SHIFT SELECT BIT INTO "C"
447 000576* 000103 ;BR IF SELECTED
448 000580* 005201 ;THIS LINE IS SELECTED & WAITING KEYBOARD ACTION
449 000602* 012760 000003 002060* 6$: INC #3,DLSTUS(R0) ;COUNT THIS LINE
450 000610* 000756 ;GO TEST NEXT LINE
451 ;
452 ;
453 ;
454 ;
455 000612* 016702 177170 ;THIS CODE STARTS UP ALL SELECTED LINES RECEIVING
456 000616* 010201 ;STRTPUP: MOV ADDR,R2 ;GET STARTING CSR ADRS TO R2
457 000620* 062702 000100 ;MOV R2,R1 ;GET STARTING CSR ADRS TO R1
458 000624* 012700 002060* ;ADD #100,R2 ;WAKE R2 LAST RCVR CSR ADRS + 10
459 000630* 005700 ;MOV #DLSTUS,R0 ;GET DL STATUS TABLE ADRS
460 000632* 001404 1$: TST (R0)+ ;SEE IF THIS LINE SELECTED
461 000634* 005761 000002 ;BRQ #25 ;BR IF NOT
462 000640* 052711 000100 ;TST #2(R1) ;FLUSH DONE BIT
463 000644* 062701 000010 2$: ADD #10,R1 ;SET RCVR INT FOR THIS LINE
464 000648* 000103 ;ADD #10,R1 ;ADVANCE TO NEXT RCVR
465 000652* 001366 ;CMP #2,R2 ;HAVE WE DONE 8 LINES?
466 ; ;BR IF NOT
467 ;
468 ;
469 ;
470 ;
471 ;
472 000654* 104407 000000* ;THIS CODE DOES THE FOLLOWING:
473 000654* 104407 000000* ;RETURNS TO MONITOR VIA "BREAK" FOR "END PASS" TIMING
474 000660* 000000* ;REPORTS ANY RECEIVER ERRORS IN THE ERROR QUEUE "EQ"
;RESTARTS ALL DL LINES WAITING TO XMIT DATA
475 ;
476 ;
477 ;
478 ;
479 ;
480 ;
481 ;
482 ;
483 ;
484 ;
485 ;
486 ;
487 ;
488 ;
489 ;
490 ;
491 ;
492 ;
493 ;
494 ;
495 ;
496 ;
497 ;
498 ;
499 ;
500 ;
501 ;
502 ;
503 ;
504 ;
505 ;
506 ;
507 ;
508 ;
509 ;
510 ;
511 ;
512 ;
513 ;
514 ;
515 ;
516 ;
517 ;
518 ;
519 ;
520 ;
521 ;
522 ;
523 ;
524 ;
525 ;
526 ;
527 ;
528 ;
529 ;
530 ;

```

```

475 000664* 005367 001106 ;COUNT EOP TIMER
476 000670* 001003 ;BNE #15 ;BR IF NOT DONE FOR "END PASS" MSG
477 000672* 005367 001102 ;DEC CNTR1 ;
478 000676* 001427 ;BRQ #25 ;STAY IN LOOP TILL CNTR1 = 0
479 000700* 004767 007742 1$: JSR #PC,RCRCK ;GO REPORT RCVR ERRORS IF ANY
480 ;
481 000704* 016700 177076 ;XSTRT: MOV ADDR,R0 ;GET BASE DL ADRS
482 000710* 062700 000004 ;ADD #4,R0 ;POINT TO XMITR CSR
483 000714* 005001 ;CLR R1 ;ZERO OFFSET
484 000716* 005761 002060* 1$: TST DLSTUS(R1) ;THIS LINE WAITING FOR XMIT?
485 000722* 005761 ;BPL #25 ;BR IF NOT
486 000724* 012765 000001 002060* ;MOV #DLSTUS(R1) ;GET THIS LINE'S XMIT INT ENABLE
487 000732* 012710 000100 2$: ADD #10,R0 ;ADVANCE TO NEXT XMITR'S CSR ADRS
488 000736* 062700 000010 ;ADD #2,R1 ;ADVANCE OFFSET FOR STATUS TABLE
489 000742* 062701 000002 ;CMP #20,R1 ;HAVE ALL LINES BEEN CHECKED
490 000746* 021701 000020 ;BNE #25 ;BR IF NOT
491 000754* 000737 ;BR #TIMR ;RETURN TO EOP TIMING
492 ;
493 ;
494 ;
495 ;
496 ;
497 ;
498 ;
499 ;
500 ;
501 ;
502 ;
503 ;
504 ;
505 ;
506 ;
507 ;
508 ;
509 ;
510 ;
511 ;
512 ;
513 ;
514 ;
515 ;
516 ;
517 ;
518 ;
519 ;
520 ;
521 ;
522 ;
523 ;
524 ;
525 ;
526 ;
527 ;
528 ;
529 ;
530 ;

```



```

643 001612* 062767 000020 000164 ADD #2,QPTR2 ;UPDATE THE QUEUE POINTER
644 001620* 022767 002160* 000156 CMP #16+20,QPTR2 ;POINTER AT HIGH LIMIT?
645 001620* 001003 BNE ;BR IF NOT
646 001630* 012767 002140* 000146 MOV #10,QPTR2 ;RESET THE POINTER
647 001640* 011000 1$: MOV (R0),R1 ;MOVE THE CSR ADRS TO R1
648 001640* 011000 MOV (R1),R2 ;SET UP R2 TO CSR CONTENTS
649 001644* 011102 MOV #300,R2 ;GET CSR CONTENTS
650 001644* 122702 000300 CMPB #300,R2 ;XMITR DOME + ENABLE MUST BE UP
651 001650* 001032 BNE ;BR IF NOT
652 001650* 005260 INC XRBPO(R0) ;ADVANCE TO NEXT DATA
653 001650* 117003 002014* MOVB XRBPO(R0),R3 ;GET NEXT DATA BYTE
654 001650* 000014 CMB #1,R3 ;IS IT THE TERMINATING CHARACTER?
655 001660* 001017 BNE ;BR IF NOT
656 001670* 016060 002040* 002014* MOV XRAD0(R0),XRBPO(R0) ;SET UP RCVR POINTER
657 001670* 062760 000003 002014* ADD #3,XRBPO(R0) ;SET RCVR POINTER TO START CODE
658 001700* 005011 CLR (R1) ;CLEAR XMITR INT ENABLE
659 001706* 005060 002120* CLR RTAB(R0) ;ZERO ERROR COUNTER THIS LINE
660 001715* 012760 000062 002060* MOV #2,DLSTUS(R0) ;INDICATE NOW RECEIVING
661 001720* 012761 000100 177774 MOV #100,-4(R1) ;SET RCVR INT ENABLE
662 001720* 110361 000002 2$: MOV #3,2(R1) ;SEND NEXT DATA BYTE
663 001730* 104400 000000 3$: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
664 001736* 005011 CLR (R1) ;DISABLE XMITR INT ENABLE IF SET
665 001740* 010160 MOV R1,CSRA ;SET UP CSR ADDRESS
666 001744* 010267 MOV R2,ACSR ;SET UP CONTENTS OF CSR
667 001750* 012767 000011 176130 MOV #11,ERRTYP ;ILLEGAL INTERRUPT
668 ***** ;*****
669 001756* 104405 000000* 000000 HDRS$BEGIN NULL ;XMITR ILLEGAL INT
670 ***** ;*****
671 001764* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
672
673 ;SOME POINTERS, VARIABLES AND CONSTANTS UNIQUE TO THIS MODULE
674
675 COUNT: 1 ;#OF 64K BREAKS TO MONITOR FOR "END PASS"
676 COUNT1: 1 ;#OF 64K BREAKS TO MONITOR FOR "TIMEOUT" TEST
677 ACTDEV: 0 ;#OF DL11S ACTIVE
678 CNTR: 0 ;COUNTS BREAKS TO MONITOR
679 CNTR1: 0 ;COUNTS GROUPS OF 64K BREAKS TO MONITOR
680 QPTR1: OPEN ;MULTI INT FIFO QUEUE POINTER-LOAD
681 QPTR2: OPEN ;MULTI INT FIFO QUEUE POINTER-UNLOAD
682 ERPTR1: OPEN ;MULTI ERROR FIFO QUEUE POINTER-LOAD
683 ERPTR2: OPEN ;MULTI ERROR FIFO QUEUE POINTER-UNLOAD
684 STATUS: 0 ;CONTAINS ADDITIONAL ERROR STATUS INFO
685
686 XRBPO: OPEN ;POINTER TO NEXT BYTE ADRS IN THE DATA
687 XRB1: OPEN ;BUFFER ON BOTH RECEIVE & TRANSMIT
688 XRB2: OPEN
689 XRB3: OPEN
690 XRB4: OPEN
691 XRB5: OPEN
692 XRB6: OPEN
693 XRB7: OPEN
694
695 002034* 002012* STATBL: STATUS ;POINTER TO ER STATUS WD
696 002036* 177777 ;PRINT OUT TERMINATOR
697
698 002040* 002320* XRAD0: XRB0 ;ADDRESSES OF THE 8 DL11
  
```

```

699 002042* 003126* XRAD1: XRB1 ;XMITR & RCVR DATA BUFFERS
700 002044* 003734* XRAD2: XRB2
701 002046* 004542* XRAD3: XRB3
702 002048* 005350* XRAD4: XRB4
703 002050* 006158* XRAD5: XRB5
704 002054* 006764* XRAD6: XRB6
705 002056* 007572* XRAD7: XRB7
706
707 ;TABLES, QUEUES & BUFFERS
708
709 DLSTUS: -BLKW 8. ;LCS SPECIFY STATUS OF EACH DL
710 RCVSW: -BLKW 8. ;LCS SPECIFY (WHEN NON-ZERO) DL HAS RECD NULLS
711 RTAB: -BLKW 8. ;RCVR ERROR COUNTERS - ONE PER DL
712 TQ: -BLKW 8. ;TMITR SERVICE FIFO QUEUE
713 EQ: -BLKW 48. ;48 WORD RCVR FIFO QUEUE
714
715 XRB0: -BLKW 195. ;390 BYTE XMIT/RCV DATA BUFFERS
716 XRB1: -BLKW 195. ;FOR 8 DL11 LINES
717 XRB2: -BLKW 195.
718 XRB3: -BLKW 195. ;THE FIRST 3 BYTES ARE NULL CHARS
719 XRB4: -BLKW 195. ;(0'S), THE REST OF EACH BUFFER WILL
720 XRB5: -BLKW 195. ;RECEIVE A START CODE OF 377 AND UP TO
721 XRB6: -BLKW 195. ;384 BYTES OF DATA, EACH DATA XFER IS
722 XRB7: -BLKW 195. ;TERMINATED WITH THE CHAR '14'.
723
724 ;JSR LINK TABLE CONSISTING OF 16 JSR'S (8 RCVR + 8 XMITR)
725 ;THAT LINK THE INTERRUPTS TO COMMON SERVICE ROUTINES
726
727 010400* 004567 170662 JSRTAB: JSR R5,RINT ;RECEIVER LINE FOR LINE 0
728 010404* 000000 ;SET UP WITH RCVR CSR ADRS
729 010406* 000000 ;IDENTIFIES THIS LINE
730 010410* 004567 171132 JSR R5,TINT ;TRANSMITTER LINE FOR LINE0
731 010414* 000000 ;SET UP WITH XMR CSR ADRS
732 010416* 000000 ;IDENTIFIES THIS LINE
733 010420* 004567 170642 JSR R5,RINT ;LINK FOR LINE 1
734 010424* 000000
735 010426* 000002 171112 JSR R5,TINT
736 010430* 004567
737 010434* 000000
738 010436* 000002
739 010440* 004567 170622 JSR R5,RINT ;LINK FOR LINE 2
740 010444* 000000
741 010446* 000004 171072 JSR R5,TINT
742 010450* 004567
743 010454* 000000
744 010456* 000004 170602 JSR R5,RINT ;LINK FOR LINE 3
745 010460* 004567
746 010464* 000000
747 010466* 004567 171052 JSR R5,TINT
748 010470* 004567
749 010474* 000000
750 010476* 000006
751 010500* 004567 170562 JSR R5,RINT ;LINK FOR LINE 4
752 010504* 000000
753 010506* 000111
754 010510* 004567 171032 JSR R5,TINT
  
```





VTAB DEC/X11 SYSTEM EXERCISER MODULE  
XVTABO.P11 12-OCT-78 12:24

MACY11 30A(1052) 12-OCT-78 17:09 PAGE 22  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0020

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

XVTABO,XVTABO/SQL/CRF:SYM=DDXCOM,XVTABO  
RUN-TIME: 1 2 .3 SECONDS  
RUN-TIME RATIO: 1774=3.8  
CORE USED: 7K (13 PAGES)