

.REM _

IDENTIFICATION

PRODUCT CODE: AC-E821E-MC
PRODUCT NAME: CXKLAE0 KL11 MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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

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

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

COPYRIGHT (C) 1973,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT:

KLA IS AN IOMOD THAT EXERCISES UP TO SIXTEEN KL11 ASYNCHRONOUS INTERFACES. IT IS CAPABLE OF EXERCISING ALL KL11 MODELS. IT USES MAINTENANCE MODE TO XMIT AND RECEIVE A BINARY COUNT PATTERN OUTPUT AND RECEIVED IN 64 CHARACTER BURSTS. THE MAJOR PORTION OF THE ERROR CHECKING IS DEFERRED TO LEVEL 0. ALL LINES SELECTED FOR TEST (UP TO 16 KL11'S WITH CONTIGUOUS ADDRESSES AND VECTORS) ARE ACTIVATED AND RUN CONCURRENTLY. ALL TRANSMIT AND RECEIVE ERRORS ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS:

HARDWARE: AT LEAST ONE KL11 INTERFACE

STORAGE:: KLA REQUIRES:
1. DECIMAL WORDS: 840
2. OCTAL WORDS: 1510
3. OCTAL BYTES: 3220

3. PASS DEFINITION:

ONE PASS OF THE KLA MODULE CONSISTS OF TRANSMITTING AND RECEIVING 8192. (TOTAL) CHARACTERS.

4. EXECUTION TIME:

VARIES WITH BAUD RATE BUT SHOULD TAKE AN AVERAGE OF ONE MINUTES TO COMPLETE ONE PASS WHEN RUNNING ALONE.

5. CONFIGURATION PARAMETERS:

DEFAULT PARAMETERS:

DVA: 174000, VCT: 300, BR1: 5, BR2: 0, DVC: 1

REQUIRED PARAMETERS:

AT CONFIGURATION TIME THE USER MUST SPECIFY:

VCT: VECTOR ADDRESS OF FIRST KL11 IF NOT 300
DVC: NO OF KL11'S IF GREATER THAN 1

6. DEVICE OPTION SETUP:

NONE REQUIRED

7. MODULE OPERATION:

7.1 TEST SEQUENCE:

- A. START: USING THE DEVICE SELECTION PARAMETER "DVIDI" THIS SECTION OF CODE SETS UP THE VECTORS OF ALL SELECTED LINES TO POINT TO THE APPROPRIATE JSR IN THE JSR LINKING TABLE.
- B. SETCSR: THIS PIECE OF CODE INSERTS THE PROPER CSR ADDRESS OF EACH ACTIVE LINE INTO THE THIRD WORD OF EACH JSR TABLE ENTRY.
- C. STUP: THIS ROUTINE INITIALIZES ALL TABLES, BUFFERS, FLAGS AND COUNTERS, THEN PROCEEDS TO TURN ON THE INTERRUPTS FOR ALL ACTIVE LINES. IT USES THE CONTENTS OF THE ACTIVE DEVICE TABLE TO FIND OUT WHICH LINES TO KICK OFF. AFTER INITIALIZING ALL LINES IT WAITS FOR COMPLETION OF 64 TRANSMITTER AND RECEIVER INTERRUPTS VIA A BREAK LOOP. IF THE 64 INTERRUPTS HAVE OCCURRED ON BOTH TRANSMITTER AND RECEIVER, OR IF THE BREAK LOOP TIMES OUT, CONTROL PASSES TO ERRCHK.

(7.1 CONT'D)

- D. TINT: THE TRANSMITTER SERVICE ROUTINE SIMPLY QUEUES UP THE REQUEST FOR SERVICE IN A FIFO QUEUE, UPDATES THE POINTER, AND RETURNS 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 SERVICING IS DONE LATER WHERE THE SERVICE CODE IS EXECUTED AT LEVEL 0.
- E. TSERV: THIS CODE RETRIEVES A POINTER FROM THE FIFO QUEUE AND BUILDS THE CSR ADDRESS. THE FOLLOWING SEQUENCE IS EXECUTED:
1. TEST FOR END OF 64. CHAR BURST - IF END EXIT - IF NOT GO TO 2
 2. TEST READY FLAG - IF NOT ASSERTED GO REPORT FALSE INTERRUPT - IF ASSERTED PROCEED TO STEP 3
 3. COUNT THE INTERRUPT FOR INDIVIDUAL LINE
 4. GENERATE AND OUTPUT NEXT CHARACTER, KEEP TRACK OF THE NUMBER OF CHARACTERS OUTPUT ON THE LINE, AND THEN EXIT BACK TO THE MONITOR.
- F. RINT: THE RECEIVER SERVICE ROUTINE STORES DATA AND STATUS INFORMATION IN A RECEIVER STARTUP TABLE, TESTS FOR THE END OF A 64. CHAR XFR SEQUENCE AND THEN EXECUTES AN "RTI". IT ALSO COUNTS RECEIVE INTERRUPTS IN A SEPARATE COUNTER FOR EACH LINE.
- G. ERRCHK: THE BULK OF THE ERROR CHECKING AND REPORTING IS DONE HERE AT THE END OF EACH 64. CHAR. BURST. THE FOLLOWING SEQUENCE IS EXECUTED:
1. TURN OFF RCVR AND XMTR INTR. ENABLES FOR ALL ACTIVE LINES
 2. SCAN THROUGH THE RECEIVER STATUS TABLE (64 ENTRIES OF TWO WORDS EACH) TO CHECK FOR AND REPORT:

(7.1, SECTION G CONT'D)

- A.) PARITY, FRAMING AND OVER-RUN ERRORS.
 - B.) RCVR FALSE INTERRUPTS
 - C.) DATA COMPARE ERRORS. ONLY IF A AND B DID NOT OCCUR.
3. CHECK RECEIVER AND TRANSMITTER INTERRUPT COUNTS FOR EACH LINE TO BE SURE THAT NO LINES WERE DROPPED OR HAD TOO MANY INTERRUPTS.
4. GO TO THE ENPS ROUTINE AFTER CHECKING ALL 64 ENTRIES.
- H. ENPS: COUNT THE 64. CHAR BURST AND TEST FOR 128. BURSTS (8192 CHARS). IF NOT END OF PASS GO TO I. IF END REPORT END OF PASS AND GO TO C.
- I. RESYNC: RESYNC THE DATA BUFFERS AND THEN RESTART AT STEP C.

7.2 DESCRIPTION OF TABLES, QUEUES, AND BUFFERS

- A. RSTAB: THIS IS A 128. WORD STATUS TABLE CONSISTING OF 64. TWO WORD ENTRIES. IT GETS LOADED DURING RECEIVER INTERRUPT SERVICE AND CHECKED AT THE END OF EACH 64. CHAR BURST. EACH ENTRY HAS THE FOLLOWING FORMAT:
- 1ST WORD: CONTENTS OF RCSR
 - 2ND WORD: LG BYTE = RCVD DATA BYTE
HI BYTE = LINE NUMBER
- B. RCNT: 16 BYTE TABLE CONTAINING AN 8 BIT INTERRUPT COUNTER FOR EACH RCVR. THE APPROPRIATE BYTE GETS INCREMENTED DURING RCVR INTR SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.
- C. TCNT: 16 BYTE TABLE CONTAINING AN 8-BIT INTERRUPT COUNTER FOR EACH TRANSMITTER. THE APPROPRIATE BYTE GETS INCREMENTED DURING DEFERRED INTR. SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.

(7.2 CONT'D)

- D. KLNT: 16 BYTE TABLE CONTAINING AN 8-BIT DATA COUNTER FOR EACH LINE. THE APPROPRIATE BYTE GETS INCREMENTED EACH TIME A CHARACTER IS TRANSMITTED ON THE LINE, AND CLEARED BEFORE THE BEGINNING OF EACH 64. WORD BURST.
- E. TQ: 16 WORD FIFO QUEUE FOR TRANSMITTER SERVICE. LOADED DURING XMTR INTERRUPT SERVICE WITH A POINTER TO THE CSR ADDRESS AND UNLOADED DURING DEFERRED XMTR SERVICE.
- F. XBUF: 16 BYTE XMTR DATA BUFFERS - ONE BYTE/XMTR
- G. RBUF: 16 BYTE RCVR DATA BUFFERS - ONE BYTE/RCVR.
- H. JSRTAB: A 128 WORD TABLE THAT CONTAINS 64 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 OVERLAVED WITH THE ADDRESS OF THE CSR FOR LINE N AND N IS THE LINE NO. IN OCTAL (00-17)

8. OPERATOR OPTIONS:

- A. LOCATION (STUP+2) CAN BE MODIFIED TO VARY THE NO. OF 64. CHAR BURSTS PER PASS.
- B. THE USER CAN USE THE "MOD" COMMAND TO DUMP THE TABLES BUFFERS DESCRIBED IN 7.2 TO OBTAIN MORE DETAILED ERROR INFORMATION.
- C. THE USER CAN MODIFY "DVID1" (KLA 14) TO SELECT OR DESELECT INDIVIDUAL KL11'S.

9. NON-STANDARD PRINTOUTS:

THERE ARE TWO ERROR PRINTOUTS WHICH SUPPLY SPECIAL INFORMATION IN THE CSRC AND STATC VALUES (CONSULT LISTING).

;KL11 A-C DEC/X11 EXERCISER MODULE

```

000000* IOMOD <KLAE > 174000,300,5,66,128.,42
000000* MODULE 140000, KLAE, 174000,300,2,66,128.,42
; TITLE KLAE DEC/X11 SYSTEM EXERCISER MODULE
; DDXCOM VERSION 6 23-NOV-78
;*****.LIST BIN*****
000000* MODNAM: .ASCII /KLAE / ;MODULE NAME.
000005* 000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGF
000006* 174000 ADDR: 174000+0 ;1ST DEVICE ADDR.
000010* 000300 VECROR: 300+0 ;1ST DEVICE VECTOR.
000011* 240 BR1: .BYTE PRTV5+0 ;1ST RR LEVEL.
000013* 000 BR2: .BYTE PRTV+0 ;2ND RR LEVEL.
000014* 000001 DVID1: +1 ;DEVICE INDICATOR 1.
000016* 000000 SR1: OPEN ;SWITCH REGISTER 1.
000017* 000000 SR2: OPEN ;SWITCH REGISTER 2.
000022* 000000 SR3: OPEN ;SWITCH REGISTER 3.
000024* 000000 SR4: OPEN ;SWITCH REGISTER 4.
;*****
000026* 140000 STAT: 140000 ;STATUS WORD.
000030* 000224 INIT: START ;MODULE START ADDR.
000032* 000224 SPOINT: MODSP ;MODULE STACK POINTER.
000034* 000000 PASCNT: 0 ;PASS COUNTER.
000036* 000200 ICOUNT: 128. ;# OF ITERATIONS PER PASS=128.
000040* 000000 ICCNT: 0 ;LCC TO COUNT ITERATIONS
000042* 000000 SDFCNT: 0 ;LCC TO SAVE TOTAL SOFT ERRORS
000044* 000000 HRDCNT: 0 ;LCC TO SAVE TOTAL HARD ERRORS
000046* 000000 SDFPAS: 0 ;LCC TO SAVE SOFT ERRORS PER PASS
000050* 000000 HRDPAS: 0 ;LCC TO SAVE HARD ERRORS PER PASS
000052* 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054* 000000 RANDSP: 0 ;RESERVED FOR MONITOR USE WHEN RAND MACRC IS CALLED
000056* 000000 CONFIG: 0 ;RESERVED FOR MONITOR USE
000060* 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000062* 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000064* 000000 SVR0: OPEN ;LCC TO SAVE R0.
000066* 000000 SVR1: OPEN ;LCC TO SAVE R1.
000068* 000000 SVR2: OPEN ;LCC TO SAVE R2.
000070* 000000 SVR3: OPEN ;LCC TO SAVE R3.
000072* 000000 SVR4: OPEN ;LCC TO SAVE R4.
000074* 000000 SVR5: OPEN ;LCC TO SAVE R5.
000076* 000000 SVR6: OPEN ;LCC TO SAVE R6.
000100* 000000 CSADR: OPEN ;ADDR OF CURRENT CSR.
000102* 000000 ACSR: OPEN ;ADDR OF GOOD DATA CR.
000104* 000000 WASADR: OPEN ;CCNTENTS OF CSR.
000106* 000000 ASTAT: OPEN ;ADDR OF BAD DATA CR.
000108* 000000 ERRPTP: OPEN ;STATUS REG CONTENTS.
000110* 000000 ASP: OPEN ;TYPE OF ERROR.
000112* 000434 AWAS: OPEN ;EXPECTED DATA.
000114* 000000 RSTRT: RESTRT ;ACTUAL DATA.
;WDTO: OPEN ;RESTART ADDRESS AFTER END OF PASS
;WRDTS: OPEN ;WRDTS TO MEMORY PER ITERATION

```

```

000116* 000000 WDFR: OPEN ;WRDTS FROM MEMORY PER ITERATION
000118* 000000 INTD: OPEN ;# OF INTERRUPTS PER ITERATION
000122* 000040 IDNUM: 42 ;MODULE IDENTIFICATION NUMBER=42
;*****.LIST BIN*****
000224* MODSP: *****
;THIS ROUTINE SETS UP THE VECTORS FOR ALL SELECTED LINES TO POINT
;TO THE APPROPRIATE JSR IN THE JSR LINK TABLE
START: MOV #64,WDIO ;64 WORDS FROM WCV/ITERATION
MOV #64,WDR ;64 WORDS FROM WCV/ITERATION
MOV #128,INTR ;128 INTERRUPTS/ITERATION
MOV VECTOR,R0 ;SET R0 TO POINT TO THE 1ST VECTOR
MOV DVID1,R1 ;LEAD R1 WITH DEVICE SELECTION PARAMETER
MOV R1,DRIVE ;INITIALIZE TO RECORD ANY LINES DROPPED
MOV JSRPTAB,R2 ;SET UP R2 TO POINT TO JSR TABLE
15: ASR R1 ;SHIFT SELECT BIT INTO "C"
BCC JS ;BR IF NOT SELECTED
MOV R2,(R0)+ ;SET UP RCVR INTR POINTER
MOV BR1,(R0)+ ;SET UP RCVR PRIORITY LEVEL
TSTB (R0)+ ;MOVE POINTER
ADD #10,R2 ;POINT R2 TO XMTB ENTRY IN JSR TABLE
MOV R2,(R0)+ ;SET UP XMTB INTR POINTER
MOV BR1,(R0)+ ;SET UP XMTB PRIORITY LEVEL
TSTB (R0)+ ;MOVE POINTER
ADD #10,R2 ;POINT R2 TO RCVR ENTRY FOR NEXT LINE
CMP #JSRPTAB+400,R2 ;IS THE POINTER AT THE END OF THE TABLE?
25: BNE JS ;BR IF NOT
RR SETCSR ;GC SET UP CSR ADDRESSES
ADD #10,R0 ;UPDATE VECTOR POINTER
ADD #10,R2 ;UPDATE JSR TABLE POINTER
BR JS ;GC CHECK FOR END OF TABLE
;THIS ROUTINE SETS UP THE JSR TABLE SUCH THAT THE APPROPRIATE
;CSR ADDRESS IS INCLUDED AS THE 3RD WORD OF EACH ENTRY
SETCSR: MOV ADDR,R0 ;GET THE FIRST CSR ADDRESS INTO R0
MOV DVID1,R1 ;LEAD R1 WITH THE DEVICE SELECTION PARAMETER
BNE JS ;NO BRANCH IF DVID1 = 0
MOV #0$,BEGIN ;
15: MOV #JSRPTAB+4,R2 ;POINT R2 TO CSR ADDRESS ENTRY
ASP R1 ;SHIFT SELECT BIT INTO "C"
BCC JS ;BR IF LINE NOT SELECTED
MOV R0,(R2) ;PUT RCVR CSR ADDRESS IN TABLE
CMP (R0)+(R0)+ ;GENERATE XMTB CSR ADDRESS IN PC
ADD #10,R2 ;POINT TO XMTB SLOT IN JSR TABLE
MOV R0,(R2) ;PUT XMTB CSR ADDRESS IN THE TABLE
CMP (R0)+(R0)+ ;GENERATE NEXT RCVR CSR ADDRESS IN R0
ADD #10,R2 ;POINT TO RCVR SLOT IN JSR TABLE
35: CMP #JSRPTAB+404,R2 ;IS POINTER BEYOND END OF TABLE?
BNE JS ;BR IF NOT
RR RESTRT ;GC SET UP ACTIVE DEVICE TABLE.
ADD #10,R0 ;UPDATE CSR ADDRESS
ADD #10,R2 ;UPDATE JSR TABLE POINTER

```

```
400 000432* 000767 BR 3$ ;GC TEST FOR END OF TABLE
401
402 ;THIS ROUTINE CLEARS BUFFERS AND TABLES, INITIALIZES FLAGS, AND STARTS
403 ;UP ALL SELECTED LINES.
404 000434* 004767 002032 RESTRT: JSR PC,CLRBUF ;GO CLEAR XMTR. AND RCVR. BUFFERS
405 000440* 004767 005062 STUP1: JSR PC,DTAB ;SET UP THE ACTIVE DEVICE TABLE.
406 000444* 004767 002040 JSR PC,CLRTAB ;GO CLEAR TABLES AND QUEUES
407 000450* 005067 002002 CLR TXCNT ;CLEAR TX TOTAL INTERRUPT COUNTER.
408 000454* 005067 002002 CLR RXCNT ;CLEAR RX TOTAL INTERRUPT COUNTER.
409 000460* 012767 001674* 001776 MOV #RSTAB,SVPTR ;INITIALIZE RCVR. STATUS TABLE POINTER
410 000466* 012767 002354* 001776 MOV #TQ,QPTR1 ;SET UP XMTR FIFO QUEUE POINTERS
411 000472* 012767 002354* 001776 MOV #TQ,QPTR2
412 000478* 012767 002354* 001776 MOV #TQ,QPTR3
413 000502* 016700 002106 1S: MOV ACTDEV,R0 ;GET COUNT OF ACTIVE DEVICES
414 000506* 016002 002574* 1S: MOV DEVTAB(R0),R2 ;GET AN ACTIVE LINE NO.
415 000512* 004767 001016 JSR PC,GETADR ;GO BUILD CSR ADDRESS IN R3
416 000516* 005763 000002 TST 2(R3) ;READ RCVR DRR TO FLUSH DCNF BIT
417 000522* 052763 000100 BIS #100,(R3) ;ENABLE RECEIVER INTERRUPTS
418 000526* 052763 000004 000004 BIS #4,(R3) ;ENABLE MAINT. MODE
419 000532* 052622 002414* 000006 INCB XBUF(R2),6(R3) ;OUTPUT CHAR ONTO TX.
420 000540* 016266 002414* 000006 MOVB XBUF(R2),6(R3) ;UP COUNT OF CHARS OUTPUT.
421 000546* 052622 002334* 000004 INCB DCNT(R2) ;COUNT CHARACTERS OUTPUT ON THIS LINE
422 000552* 052763 000100 000004 BIS #100,(R3) ;ENABLE TX INTERRUPTS.
423 000556* 005300 000100 DEC R0 ;COUNT ONE KICKED OFF
424 000562* 052622 001704 TST R0 ;GO TEST FOR NEXT ONE
425 000566* 012767 000006 001664 MOV #R,CNTR ;INITIALIZE COUNTER TO WAIT AT LEAST
426 000576* 005004 10S: CLR R4 ;30 SECONDS BEFORE TIMING OUT
427 2S:
428 000600* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
429 000604* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
430 000610* 122767 000100 001641 CMPB #64, TXCNT+1 ;64 TRANSMITTER INTERRUPTS?
431 000616* 103044 000100 001632 BCT #64, RXCNT ;YES - BRANCH TO WAIT
432 000622* 022767 000100 001632 CMP #64, RXCNT ;YES - 64 RECEIVER INTERRUPTS?
433 000626* 003405 3S: BLE AS ;YES - GO CHECK FOR ERRORS
434 000632* 005304 3S: DEC #2 ;TIMEOUT?
435 000636* 005367 001622 BNE CNTR ;WAIT SOME MORE
436 000640* 001356 BNE 10S ;EACH PASS OF THE SMALL LOOP TAKES
437 000644* 000167 000272 4S: JMP ERRCHK ;AT LEAST 5 SECONDS
438 ;BRANCH IF NOT DONE WITH 6 PASSES OF
439 ;THE SMALL COUNTER
440 000646* 010577 001614 ;TRANSMITTER INTERRUPT SERVICE - ENTERED VIA APPROPRIATE JSR TABLE
441 ;ENTRY WITH R5 POINTING TO THE CSR ADDRESS - CONTENTS OF R5
442 ;GETS QUEUED UP IN FIFO QUEUE AND ROUTINE RETURNS CONTROL BACK TO
443 ;THE MONITOR VIA A PIRQ TO DEFER SERVICING XMTR AT LEVEL 0
444 000652* 010577 001614 TINT: MOV #R,QPTR1 ;STORE CONTENTS OF R5 IN THE QUEUE
445 000656* 022767 002414* 001600 ADD #TQ,QPTR1 ;UPDATE THE QUEUE POINTER
446 000662* 001003 001600 CMP #TQ+40,QPTR1 ;POINTER AT END OF QUEUE?
447 000666* 001003 001570 BNE IS ;IF NOT
448 000672* 012767 002354* 001570 MOV #TQ,QPTR1 ;SET THE POINTER
449 000676* 012605 1S: MOV (R6)+,R5 ;RESTORE THE OTHER GUY'S R5
450 ;-----
451 000700* 000004 000000* 000706* PIRQS,BEGIN,TSERV ; QUEUE UP TO CONTINUE AT TSERV AND RTI
```

```
456 ;-----
457 ;DEFERRED XMTR SERVICE - THIS ROUTINE RETRIEVES POINTER TO CSR ADDRESS
458 ;FROM THE FIFO QUEUE AND SERVICES THE LINE AT LEVEL 0
459 000706* 017700 001556 TSERV: MOV #QPTR2,R0 ;GET POINTER FROM THE QUEUE
460 000712* 062767 000002 001550 ADD #2,QPTR2 ;UPDATE THE QUEUE POINTER
461 000720* 022767 002414* 001542 CMP #TQ+40,QPTR2 ;POINTER AT HIGH LIMIT
462 000726* 001003 1S: BNE IS ;IF NOT
463 000732* 012767 002354* 001532 MOV #TQ,QPTR2 ;RESET THE POINTER
464 000736* 012001 1S: MOV (R0),R1 ;MOVE CSR ADDRESS INTO R1
465 000740* 011000 1S: MOV #R0,R0 ;MOVE LINE # INTO R0
466 000744* 010577 001511* 1S: INCB TXCNT+1 ;COUNT TOTAL TX INTERRUPTS.
467 000748* 0105260 002314* 1S: INCB TCNT(R0) ;COUNT THE INTERRUPT
468 000752* 010511 1S: TSTB (R1) ;XMTR READY FLAG ASSERTED?
469 000756* 000011 1S: BPL AS ;IF NOT
470 000760* 001427 000100 001472 BPL CMPB #64, TXCNT ;64 CHARACTERS TRANSMITTED?
471 000764* 0105260 002414* 000002 BEQ #64, R0 ;YES - BRANCH TO EXIT
472 000768* 0105260 002414* 000002 INCB XBUF(R0) ;GENERATE NEXT DATA BYTE
473 000772* 0105260 002414* 000002 MOVB XBUF(R0),2(R1) ;LOAD THE XMTR BUFFER
474 000776* 0105260 002334* 000002 INCB TXCNT ;UP TOTAL COUNT OF CHARS OUTPUT.
475 000780* 0105260 002334* 000002 INCB DCNT(R0) ;COUNT CHARACTERS OUTPUT ON THIS LINE
476 000784* 0105260 000000 4S: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
477 000788* 0105260 000000 4S: MOV #R1,CSRA ;SAVE CSR ADDRESS
478 000792* 011167 000000 4S: MOV (R1),ACSR ;SAVE CONTENTS OF THE CSR
479 000796* 011167 000000 4S: BICB #100,(R1) ;DISABLE XMTR INTERRUPT
480 000800* 012767 000011 177050 MOV #R1,ERRTP ;ILLEGAL INTERRUPT
481 ;*****
482 ;RDERS,BEGIN,NULL ;XMTR FALSE INTERRUPT
483 ;*****
484 001036* 010405 000000* 000000 5S: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
485 001044* 010440 000000*
486 001048*
487 ;RECEIVER INTERRUPT SERVICE-ENTERED VIA APPROPRIATE JSR TABLE ENTRY
488 ;STORES PERTINENT INFORMATION IN THE RECEIVER STATUS TABLE THAT WILL
489 ;BE CHECKED IF 64 CHARACTERS HAVE BEEN RECEIVED
490 001050* 010046 000100 001376 RINT: MOV R0,-(R6) ;SAVE R0 AND R1 ON THE STACK
491 001054* 010146 000100 001376 MOV R1,-(R6)
492 001058* 003414 000100 001376 CMP #64, RXCNT ;64 CHAR'S RECEIVED?
493 001062* 016700 001374 BLE AS ;YES - BRANCH
494 001066* 011501 001374 MOV SVPTR,R0 ;INC - GET RCVR STATUS TABLE POINTER
495 001070* 011501 001374 MOV (R5),R1 ;GET RCVR CSR ADDRESS
496 001074* 011120 000002 001374 MOV (R1),(R0) ;SAVE THE RCVR CONTENTS
497 001078* 011120 000002 001374 MOVB 2(R1),(R0)+ ;SAVE THE PCSR CONTENTS
498 001082* 011120 000002 001374 MOV (R1),3(R0) ;REREAD CSR IN CASE ERROR SET BETWEEN THE
499 001100* 051160 177775 1S: MOVB 2(R5),(R0)+ ;LAST TWO INSTRUCTIONS
500 001104* 010067 000002 001374 MOV R0,SVPTR ;SAVE THE LINE NUMBER
501 001108* 010605 000002 001374 MOV 2(R5),R5 ;GET LINE NO. INTO R5
502 001112* 0105260 002274* 1S: INCB RXCNT ;COUNT THE INTERRUPT FROM THIS LINE
503 001116* 005267 001330 2S: INC RXCNT ;INCREMENT RX INTERRUPT TOTAL COUNT.
504 001120* 012600 001330 2S: MOV (R6)+,R1 ;RESTORE THE OTHER GUY'S REGISTER
505 001124* 012600 001330 2S: MOV (R6)+,R5
506 001128* 000002 RTI ;RETURN CONTROL BACK TO OTHER GUY
507 ;RECEIVER ERROR CHECKING AND CLEANUP ROUTINES
508
509
510
511
```



```

512
513
514 001140 016700 001450 ;THIS ROUTINE DISABLES INTERRUPTS FROM ALL ACTIVE LINES
ERRCHK: MOV ACTDEV,R0 ;GET COUNT OF ACTIVE DEVICES
515 001144 016002 002574 IS: MOV DEVTAB(R0),R2 ;GET ACTIVE LINE NO.
516 001150 004767 000360 JSR PC,GETADR ;GC BUILD ADDRESS IN R3
517 001154 042713 000190 BIC #10,(R3) ;TURN OFF RECEIVER
518 001160 042763 000100 000004 BIC #100,(R3) ;TURN OFF TRANSMITTER.
519 001166 005300 ;COUNT ONE GUY OFF
520 001170 100365 BPL R5 ;RR TIL ALL CFF

;THIS ROUTINE SCANS THROUGH THE 64 ENTRY RECEIVER STATUS TABLE
;CHECKING FOR AND REPORTING ANY ERRORS.
523 CHK1: CMP RXCNT,#64. ;MAKE SURE COUNT IS NO LARGER THAN TABLE
524 BNE #64,RXCNT ;
525 001200 003403 ;
526 001202 012767 000100 001250 1S: MOV #RSTAB,R1 ;GET STATUS TABLE POINTER
527 001210 012701 001674 2S: CLR R0 ;INDICATE NO HARDWARE FAILURES YFT.
528 001214 005000 ;GET LINE NO. INTO R2
529 001216 018102 000003 4S: MOVVB 3(R1),R2 ;POINT TO LO BYTE OF CSR
530 001222 105711 ;BFI IF DCNE WAS SET
531 001224 100410 JSR PC,RCVERR ;SETUP FOR ERROR REPORT
532 001228 004767 JSR PC,RCRTP ;ILLEGAL INTERRUPT
533 001232 012767 000011 176646 MOV ;*****RECEIVER FALSE INTERRUPT*****
534 HRDERS,BEGIN,NULL ;*****RECEIVER FALSE INTERRUPT*****
535 001240 104405 000000 000000 5S: INCB RBUF(R2) ;BUMP EXPECTED DATA
536 001252 005700 TST R0 ;HARDWARE ERRORS?
537 001254 001014 BNE #340,RBUF(R2) ;DO NOT REPORT DATA ERRORS THEN
538 001256 142762 000340 002434 BICB #340,RBUF(R2) ;MASK OFF BITS <7:5> TO CHECK ONLY
539 001264 142761 000340 000002 BICB #340,2(R1) ;FIVE BITS ON
540 001272 126261 002434 000002 CMPB RBUF(R2),2(R1) ;DID RCVD DATA CHECK OK?
541 001300 001402 BEQ #68 ;BR IF YES
542 001302 004767 JSR PC,DATBAD ;GC REPORT DATA ERROR
543 001306 022121 000244 6S: CMP (R1)+,(R1)+ ;POINT R1 TO NEXT TABLE ENTRY
544 001310 005367 001144 DEC RXCNT ;ALL CHARS RECEIVED CHECKED?
545 001314 001337 BNE #2S ;INC- GO CHECK NEXT ENTPY

;THIS ROUTINE REPORTS ANY LINE RECEIVING AN INCORRECT NUMBER OF INTERRUPTS
548 CKLINS: MOV ACTDEV,R1 ;GET ACTIVE DEVICE COUNT
549 001322 116102 002574 3S: MOV DEVTAB(R1),R2 ;GET ACTIVE DEVICE LINE NO.
550 001324 002274 002334 4S: CMP RCNT(R2),DCNT(R2) ;CORRECT NUMBER OF RCVR INTERRUPTS?
551 001334 001402 BEQ #4S ;BR IF YES
552 001336 004767 000064 JSR PC,BADR ;GC REPORT BAD RCVR
553 001340 126262 002314 002334 4S: CMPB TCNT(R2),DCNT(R2) ;CORRECT NUMBER OF XMTF INTERRUPTS?
554 001342 001402 BEQ #4S ;BR IF YES
555 001352 004767 000112 5S: JSR PC,BADT ;GC REPORT BAD XMTF
556 001356 005301 DEC R1 ;COUNT ONE GUY CHECKED
557 001360 100360 BPL R5 ;BR TIL ALL CHECKED
558 001362 000530 BR ENPS ;GC CHECK FOR END OF PASS

561 001364 001370 RING: MRING
562 001366 177777 -1
563
564 001370 051045 047111 020107 MRING: .ASCIZ /RING SET- BAD LINE DROPPED%/
565 001376 042523 026524 041040

```

```

568 001404 042101 046040 047111
569 001412 020105 051104 050117
570 001420 042520 022504 000
571 001426 001426 .EVEN
572
573 ;ROUTINE TO REPORT BAD LINES (TOO MANY OR TOO FEW INTERRUPTS)
574 BADR: JSR PC,GETADR ;GC BUILD CSF ADDRESS
575 001432 010367 176442 MOV R3,CSRA ;SAVE CSF ADDRESS
576 001436 116267 002334 176436 MOV DCNT(R2),ACSR ;CHARACTERS XMTD
577 001444 116267 002274 176432 MOVVB RCNT(R2),ASTAT ;# OF RCVR INTERRUPTS
578
579 MOV #14,BRTPY ;
580 001452 012767 000014 176426 ;*****RECEIVER FALSE INTERRUPT*****
581 HRDERS,BEGIN,NULL ;INCORRECT NUMBER OF RCVR INTERRUPTS
582 ;*****RECEIVER FALSE INTERRUPT*****
583 ;NOTE THAT CSRC VALUE IS # OF CHARACTERS
584 ;TRANSMITTED, AND STATC VALUE IS # OF
585 ;RECEIVER INTERRUPTS
586
587 001466 000207 RTS PC ;RETURN TO CALLER
588
589 001470 004767 000040 BADT: JSR PC,GETADR ;GC BUILD CSF ADDRESS
590 001474 022323 176376 CMP (R3)+(R3)+ ;MAKE IT A XMTF CSR ADDRESS
591 001478 010367 002334 176372 MOV R3,CSRA ;SAVE CSF ADDRESS
592 001502 116267 002334 176372 MOVVB DCNT(R2),ACSR ;CHARACTERS XMTD
593 001510 116267 002314 176366 MOVVB TCNT(R2),ASTAT ;# OF XMTF INTERRUPTS
594
595 MOV #14,BRTPY ;
596 001516 012767 000014 176362 ;*****RECEIVER FALSE INTERRUPT*****
597 HRDERS,BEGIN,NULL ;INCORRECT NUMBER OF XMTF INTERRUPTS
598 ;*****RECEIVER FALSE INTERRUPT*****
599 ;NOTE THAT CSRC VALUE IS # OF CHARACTERS
600 ;TRANSMITTED, AND STATC VALUE IS # OF
601 ;TRANSMITTER INTERRUPTS
602
603 001532 000207 RTS PC ;RETURN TO CALLER
604
605 001534 010203 GETADR: MOV R2,R3 ;GET LINE NO.
606 001536 006303 ASL R3 ;BUILD CSP ADDRESS
607 001540 006303 ASL R3
608 001544 006303 ASL R3
609 001544 006703 ADD ADDR,R3
610 001550 000207 RTS PC ;RETURN TO CALLER
611
612 ;ROUTINE TO REPORT RCVR DATA COMPARE ERRORS
613 DATBAD: JSR PC,GETADR ;GC BUILD CSF ADDRESS
614 001556 010367 176316 MOV R3,CSRA ;SAVE CSF ADDRESS
615 001560 116167 000002 176320 MOVVB 2(R1),AWAS ;SAVE BAD DATA
616 001570 005721 TST R1 ;GENERATE RCVR DATA ADDRESS
617 001572 010167 176306 MOV R1,WASADR ;SAVE ADDRESS OF BAD DATA
618 001576 005741 TST -(R1) ;RESET R1
619 001600 012702 002434 MOV #RBUF,R5 ;GENERATE ADDRESS OF GOOD DATA
620 001604 060205 ADD R2,R5
621 001606 111567 176274 MOVVB (R5),ASR ;SAVE GOOD DATA
622 001612 010567 176264 MOV R2,ASADR ;SAVE ADDRESS OF GOOD DATA
623 ;*****

```

```

624 001616* 104404 000000* DATES,BEGIN ;DATA ERROR!!!
625 ***** ;*****
626 001622* 000207 RTS PC ;RETURN TO CALLER
627
628 ;ROUTINE TO SETUP FOR RECEIVER ERROR PRINTOUTS
629 RCVERR: INC R0 ;INDICATE HARDWARE ERROR.
630 JSR PC,GETADR ;GO BUILD CSR ADDRESS
631 MOV R3,CSRA ;STUFF IT IN CSRA
632 MOV (R1),ACSR ;GET CONTENTS IN ACSR
633 RTS PC ;RETURN TO CALLER
634
635 ;THIS ROUTINE CHECKS FOR AND REPORTS END OF PASS
636 ENPS: ENDITS,BEGIN ;SIGNAL END OF ITERATION.
637 ;MONITOR SHALL TEST END OF PASS
638
639 ;THIS ROUTINE RESTARTS EACH 64 CHAR XFR SEQUENCE
640 RESYNC: MOV #RBUF,R0 ;RESYNC DATA FOR NEXT PASS
641 MOV #RBUF,R1
642 MOV R0,(R1)+
643 IS: CMP #RBUF+20,R0 ;DONE 16 BYTFS?
644 BNE IS ;BR IF NOT
645 JMP STUP1 ;RESUME.
646
647 ;TABLES AND BUFFERS
648 RSTAB: .BLKW 128. ;128 WORD(64 ENTRIES)RCVR STATUS TABLE
649 RCNT: .BLKW 8. ;RCVR INTERRUPT COUNTERS
650 TCNT: .BLKW 8. ;XMTR INTERRUPT COUNTERS
651 DCNT: .BLKW 8. ;CHARACTER COUNTERS
652
653 TQ: .BLKW 16. ;16 WORD XMTR FIFO QUEUE
654
655 XBUF: .BLKW 8. ;16 BYTE XMTR DATA BUFFERS
656 RBUF: .BLKW 8. ;16 BYTE RCVR DATA BUFFERS
657
658 ;POINTERS, CONSTANTS, AND VARIABLES
659 COUNT: OPEN ;END OF PASS COUNTER
660 TXCNT: OPEN ;TX TOTAL INTERRUPTS COUNTER (HIGH BYTE).
661 RXCNT: OPEN ;RX TOTAL INTERRUPTS COUNTER.
662 CNTR: OPEN ;BREAK LOOP COUNTER
663 SVPTR: OPEN ;TEMP STORAGE FOR RSTAB POINTER
664 QPTR1: OPEN ;XMTR FIFO QUEUE POINTER - LOAD
665 QPTR2: OPEN ;XMTR FIFO QUEUE POINTER - UMLCAD
666
667 ;SUBROUTINE TO CLEAR DATA BUFFERS AT BEGINING OF EACH NEW PASS
668 CLRBUF: MOV #RBUF,R0 ;SET UP R0 TO POINT TO BEGINING
669 IS: CLR (R0)+ ;CLEAR A WORD
670 CMP #RBUF+20,R0 ;END OF RCVR BUFFER?
671 BNE IS ;BR TIL ALL CLEAR
672 RTS PC ;RETURN TO CALLER
673
674 ;SUBROUTINE TO CLEAR TABLES AND QUFUES
675

```

```

680 002510* 012700 001674* CLRTAB: MOV #RSTAB,R0 ;SET UP R0 TO POINT TO BEGINING
681 002514* 005020 IS: CLR (R0)+ ;CLEAR A WORD
682 002516* 022700 002414* CMP #TQ+40,R0 ;END?
683 002524* 000207 BNE IS ;BR IF NOT
684 RTS PC ;RETURN TO CALLER
685
686 ;THIS ROUTINE SETS UP AN ACTIVE DEVICE TABLE TO REMEMBER HOW MANY
687 ;AND WHICH LINES WERE ACTIVE DURING TEST - IT IS USED DURING THE
688 ;ERROR CHECKING ROUTINES FOR VARIOUS PURPOSES
689 DTAB: CLR R0 ;SET UP R0 AS TOTAL LINE COUNTER
690 CLR R1 ;INITIALLY SET TO MINUS ONE
691 CLR R2 ;INITIALLY SET TO MINUS ONE
692 MOV R1,R3 ;GET DEVICE SELECTION PARAMETER
693 INC R0 ;COUNT ONE DEVICE
694 IS: INC R0,R0 ;16 LINES CHECKED?
695 CMP #20,R0 ;BR IF NOT
696 BNE IS ;SAVE THE COUNT OF ACTIVE LINES
697 MOV R1,ACTDEV ;RETURN TO CALLER
698 RTS PC ;SET SELECT BIT INTO "C"
699 2$: ASR R2 ;BR IF NOT SELECTED
700 BCC IS ;COUNT ACTIVE LINE
701 INC R1 ;STORE ACTIVE LINE NO. IN THE TABLE
702 MOV R0,DEVTAB(R1) ;STORE NEXT LINE
703 BR IS
704
705 DEVTAB: .BLKW 8. ;16 BYTE ACTIVE DEVICE TABLE
706 ACTDEV: OPEN ;STORES COUNT OF NO. OF ACTIVE LINES MINUS ONE
707 DVICE: OPEN ;DEVICE SELECTION INDICATOR
708
709 ;JSR LINK TABLE CONSISTING OF 32 JSR'S (16 RCVR AND 16 XMTR) THAT
710 ;LINK THE INTERRUPTS TO THE COMMON SERVICE ROUTINES
711 JSRTAB: JSR R5,RINT ;RECEIVER LINK FOR LINE 0
712 0 ;SET UP WITH RCVR CSR ADDRESS
713 0 ;LINE NUMBER
714 JSR R5,TINT ;XMTR LINK FOR LINE 0
715 0 ;SET UP WITH XMTR CSR ADDRESS
716 0 ;LINE NUMBER
717 JSR R5,RINT ;LINK FOR LINE 1
718 0
719 1 JSR R5,TINT
720 0
721 1 JSR R5,RINT ;LINK FOR LINE 2
722 0
723 2 JSR R5,RINT
724 0
725 2 JSR R5,TINT
726 0
727 3 JSR R5,RINT ;LINK FOR LINE 3
728 0
729 3 JSR R5,TINT
730 0
731 3 JSR R5,RINT ;LINK FOR LINE 4
732 0
733 3 JSR R5,TINT
734 0
735 3 JSR R5,RINT ;LINK FOR LINE 4

```

736	002774*	000000		0		
737	002776*	000004		4		
738	002730*	004567	175712	JSR	R5,TINT	
739	002734*	000000		0		
740	002736*	000004		4		
741	002740*	004567	176104	JSR	R5,RINT	;LINK FOR LINE 5
742	002744*	000000		0		
743	002746*	000005		5		
744	002750*	004567	175672	JSR	R5,TINT	
745	002754*	000000		0		
746	002756*	000005		5		
747	002760*	004567	176064	JSR	R5,RINT	;LINK FOR LINE 6
748	002764*	000000		0		
749	002766*	000006		6		
750	002770*	004567	175652	JSR	R5,TINT	
751	002774*	000000		0		
752	002776*	000006		6		
753	003000*	004567	176044	JSR	R5,RINT	;LINK FOR LINE 7
754	003004*	000000		0		
755	003006*	000007		7		
756	003014*	004567	175632	JSR	R5,TINT	
757	003014*	000000		0		
758	003016*	000007		7		
759	003020*	004567	176024	JSR	R5,RINT	;LINK FOR LINE 10
760	003024*	000000		0		
761	003026*	000010		10		
762	003030*	004567	175612	JSR	R5,TINT	
763	003034*	000000		0		
764	003036*	000010		10		
765	003040*	004567	176004	JSR	R5,RINT	;LINK FOR LINE 11
766	003044*	000000		0		
767	003046*	000011		11		
768	003050*	004567	175572	JSR	R5,TINT	
769	003054*	000000		0		
770	003056*	000011		11		
771	003060*	004567	175764	JSR	R5,RINT	;LINK FOR LINE 12
772	003064*	000000		0		
773	003066*	000012		12		
774	003070*	004567	175552	JSR	R5,TINT	
775	003074*	000000		0		
776	003076*	000012		12		
777	003100*	004567	175744	JSR	R5,RINT	;LINK FOR LINE 13
778	003104*	000000		0		
779	003106*	000013		13		
780	003110*	004567	175532	JSR	R5,TINT	
781	003114*	000000		0		
782	003116*	000013		13		
783	003120*	004567	175724	JSR	R5,RINT	;LINK FOR LINE 14
784	003124*	000000		0		
785	003126*	000014		14		
786	003130*	004567	175512	JSR	R5,TINT	
787	003134*	000000		0		
788	003136*	000014		14		
789	003140*	004567	175704	JSR	R5,RINT	;LINK FOR LINE 15
790	003144*	000000		0		
791	003146*	000015		15		

792	003150*	004567	175472	JSR	R5,TINT	
793	003154*	000000		0		
794	003156*	000015		15		
795	003160*	004567	175664	JSR	R5,RINT	;LINK FOR LINE 16
796	003164*	000000		0		
797	003166*	000016		16		
798	003170*	004567	175452	JSR	R5,TINT	
799	003174*	000000		0		
800	003176*	000016		16		
801	003200*	004567	175644	JSR	R5,RINT	;LINK FOR LINE 17
802	003204*	000000		0		
803	003206*	000017		17		
804	003210*	004567	175432	JSR	R5,TINT	
805	003214*	000000		0		
806	003216*	000017		17		
807						
808		000001				

.END

SETCSR	000344R	375																		
SOPCNT	000042R	318#	382#																	
SOPERS =	104406	354#																		
SOPERS	000046R	320#																		
SPOINT	000032R	314#																		
SPSIZ =	000040	1#	347																	
SR1	000016R	307#																		
SR2	000020R	308#																		
SR3	000022R	309#																		
SR4	000024R	310#																		
START	000248R	313#	356#																	
STAT	000268R	312#																		
STUP1	000440R	406#	646																	
SVPTR	002464R	410*	495	502*	667#															
SVR0	000062R	327#																		
SVR1	000064R	328#																		
SVR2	000066R	329#																		
SVR3	000070R	330#																		
SVR4	000072R	331#																		
SVR5	000074R	332#																		
SVR6	000076R	333#																		
SYSCNT	000052R	322#																		
TCNT	002314R	468#	555	593	652#															
TINT	000646R	448#	714	720	732	738	744	750	756	762	768	774	780							
		786	792	798	804															
TQ	002354R	411	412	450	452	462	464	655#	682											
TRPDFD =	000022	354#																		
TSERV	000766R	455#	460#																	
TXCNT	002456R	408*	421*	432	467*	471	475*	663#												
VECTOR	000010R	303#																		
WASADR	000104R	337#																		
WDFR	000116R	344#																		
WDTD	000114R	343#																		
XBUF	002414R	419#	420	473*	474	642	657#	673												
XFLAG	000005R	301#																		
	= 003220R	571#	650#	651#	652#	653#	655#	657#	658#	705#										

- ABS. 000000 000
 003220 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 XKLAE0,XKLAE0/SOL/CRF:SYM=DDXCON,XKLAE0
 RUN-TIME: 1 2 3 SECONDS
 RUN-TIME RATIO: 3074=6.2
 CORE USED: 7K (13 PAGES)