

.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 WRITE AND IN RECEIVE A BINARY COUNT PATTERN OUTPUT AND RECEIVED IN 64 CHARACTER BURSTS. TO THE MAJOR PORTION OF THE ERROR CHECKING IS DEFERRED. 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:

VARIABLES WITH BAUD RATE BUT SHOULD TAKE AN AVERAGE OF ONE MINUTE 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 NCT 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
"DIVID" 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 PRO-
CEEDS TO TURN ON THE INTERRUPTS FOR ALL
OF THE ACTIVE LINES. IT USES THE CONTENTS
OF THE ACTIVE DEVICE TABLE TO FIND OUT
WHICH LINES TO KICK OFF. AFTER INITIAL-
IALIZING 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 BRCHK.

(7.1 CONT'D)

SEP 0004

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 WITH
IN THE QUEUE IS A POINTER TO THE INTER-
RUPTING CSR ADDRESS. THE ACTUAL SERVICE
VICING 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
TO 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 RE-
CEIVER STARTUP TABLE. TESTS FOR THE
END OF A 64 CHAR XFR SEQUENCE AND
THEN EXECUTES AN "RPI". IT ALSO COUNTS
RECEIVE INTERRUPTS IN A SEPARATE
COUNTER FOR EACH LINE.

G. ERCHK:

- THE BULK OF THE ERROR CHECKING AND RE-
PORTING 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: LO 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 DEERRD 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 OVERLAPPED WITH THE ADDRESS OF THE CSR FOR LINE N AND N IS THE LINE NO. IN OCTAL (00-17)

- 8. OPERATOR OPTIONS:

 - A. LOCATION (STUR*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 "OVID1" (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,174000,300,5,66,128,,42
; TITLE KLAE DEC/X11 SYSTEM EXERCISER MODULE
; BDXCOM VERSION 6 23-NOV-78
;*****.LIST BIN
000000- BEGIN:
000000- MODNAM: .ASCII /KLAE / ;MODULE NAME
000005- XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGF
000006- ADDR: 174000+0 ;1ST DEVICE ADDR.
000010- VECTOR: 300+0 ;1ST DEVICE VECTOR.
000012- BR1: .BYTE PRTY5+0 ;1ST BR LEVEL.
000013- BR2: .BYTE PRTY+0 ;2ND BR LEVEL.
000014- SR1: .+1 ;DEVICE INDICATOR 1.
000016- SR2: .OPEN ;SWITCH REGISTER 1
000020- SR3: .OPEN ;SWITCH REGISTER 2
000022- SR4: .OPEN ;SWITCH REGISTER 3
000024- SR5: .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 ICNT: 128. ;# OF ITERATIONS PER PASS=128.
000040- 000000 ICCNT: 0 ;LOC TO COUNT ITERATIONS
000042- 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044- 000000 SOFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046- 000000 HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050- 000000 SVSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052- 000000 RANNUM: 0 ;# OF SYS ERRORS ACCUMULATED
000054- 000000 CONFIG: 0 ;RESERVED FOR MONITOR USE
000056- 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000058- 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062- 000000 SVR0: .OPEN ;LOC TO SAVE R0.
000064- 000000 SVR1: .OPEN ;LOC TO SAVE R1.
000066- 000000 SVR2: .OPEN ;LOC TO SAVE R2.
000068- 000000 SVR3: .OPEN ;LOC TO SAVE R3.
000072- 000000 SVR4: .OPEN ;LOC TO SAVE R4.
000074- 000000 SVR5: .OPEN ;LOC TO SAVE R5.
000076- 000000 SVR6: .OPEN ;LOC TO SAVE R6.
000100- 000000 CSRA: .OPEN ;ADDR OF CURRENT CSF.
000102- 000000 SBADR: .OPEN ;ADDR OF GOOD DATA, OR
000104- 000000 ACSR: .OPEN ;CCNTENTS OF CSR.
000106- 000000 WASADR: .OPEN ;ADDR OF BAD DATA, CR
000108- 000000 ERRTYP: .OPEN ;STATUS REG CONTENTS.
000110- 000000 ASR: .OPEN ;TYPE OF ERROR.
000112- 000434 AMAS: .OPEN ;EXPECTED DATA.
000114- 000000 RSTR: .RSTRRT ;ACTUAL DATA.
;WDTO: .OPEN ;RESTART ADDRESS AFTER END OF PASS
;WRD5: .OPEN ;WRD5 TO MEMORY PER ITERATION
```

```
000116- 000000 WDFR: .OPEN ;WORDS FROM MEMORY PER ITERATION
000120- 000000 INTR: .OPEN ;# OF INTERRUPTS PER ITERATION
000122- 000042 IDNUM: .WORD 42 ;MODULE IDENTIFICATION NUMBER=42
;*****.LIST ENDR
;*****
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,WDTO ;64 WORDS TO MEM/ITERATION
MOV #64,WDFR ;64 WORDS FROM MEM/ITERATION
MOV #128,INTR ;128 INTERRUPTS/ITERATION
MOV VECTOR,R0 ;SET R0 TO POINT TO THE 1ST VECTOR
MOV DVID1,R1 ;LOAD R1 WITH DEVICE SELECTION PARAMETER
MOV #1,DEVICE ;INITIALIZE TO RECORD ANY LINES DROPPED
MOV #JSRTAB,R2 ;SET UP R2 TO POINT TO JSR TABLE
1$: ASR R1 ;SHIFT SELECT BIT INTO "C"
BCC JS ;BR IF NOT SELECTED
MOV R2,(R0)+ ;SET UP RCVR INTR POINTER
TSTB (R0)+ ;MOVE POINTER
ADD #1,R2 ;POINT R2 TO XMTF ENTRY IN JSR TABLE
MOV R2,(R0)+ ;SET UP XMTF INTR POINTER
TSTB (R0)+ ;MOVE POINTER
ADD #1,R2 ;POINT R2 TO RCVR ENTRY FOR NEXT LINE
2$: CMP #JSRTAB+400,R2 ;IS THE POINTER AT THE END OF THE TABLE?
BNE JS ;BR IF NOT
3$: SETCSR ;GC SET UP CSR ADDRESSES
ADD #10,R0 ;UPDATE VECTOR POINTER
ADD #20,R2 ;UPDATE JSR TABLE POINTER
BR JS ;GC CHECK FOR END OF TABLE
;*****
;THIS ROUTINE SETS UP THE JSR TABLE SUCH THAT THE APPROPRIATE
;CSR ADDRESS IS INCLUDED AS THE 3RD WORD OF EACH ENTRY
SETCSR: MOV ADDR,R0 ;GET THE FIRST CSR ADDRESS INTO R0
MOV #DVID1,R1 ;LOAD R1 WITH THE DEVICE SELECTION PARAMETER
BNE JS ;NO BRANCH IF DVID1 = 0
;*****
1$: MOV #JSRTAB+4,R2 ;POINT R2 TO CSR ADDRESS ENTRY
2$: ASR R1 ;SHIFT SELECT BIT INTO "C"
BCC JS ;BR IF LINE NOT SELECTED
MOV R2,(R2) ;PUT RCVR CSR ADDRESS IN PC
CMP (R0)+(R0)+ ;GENERATE XMTF CSR ADDRESS IN PC
ADD #10,R2 ;POINT TO XMTF SLOT IN JSR TABLE
MOV R2,(R0)+ ;PUT XMTF CSR ADDRESS IN THE TABLE
3$: ADD #10,R2 ;GENERATE NEXT RCVR CSR ADDRESS IN PC
CMP #JSRTAB+404,R2 ;IS POINTER BEYOND END OF TABLE?
BNE JS ;BR IF NOT
4$: BR RSTRRT ;GC SET UP ACTIVE DEVICE TABLE.
ADD #10,R0 ;UPDATE CSR ADDRESS
ADD #20,R2 ;UPDATE JSR TABLE POINTER
```

```
400 000432* 000767 BR 3S ;GC TEST FOR END OF TABLE
401
402 ;THIS ROUTINE CLEARS BUFFERS AND TABLES, INITIALIZES FLAGS, AND STARTS
403 ;UP ALL SELECTED LINES.
404
405 000434* 004767 002032 JSR PC,CLRRBUF ;GO CLEAR XMTR. AND RCVR. BUFFERS
406 000440* 004767 002062 JSR PC,DTAB ;SET UP THE ACTIVE DEVICE TABLE.
407 000444* 004767 002040 JSR PC,CLRTAB ;GO CLEAR TABLES AND QUEUES
408 000450* 005067 002002 CLR TXCNT ;CLEAR TX TOTAL INTERRUPT COUNTER.
409 000454* 005067 002000 CLR RXCNT ;CLEAR RX TOTAL INTERRUPT COUNTER.
410 000460* 012767 001674* 001776 MOV #RSTAB,SVPTR ;INITIALIZE RCVR STATUS TABLE POINTER
411 000466* 012767 002354* 001772 MOV #TQ,QPTR1 ;SET UP XMTR FIFO QUEUE POINTERS
412 000474* 012767 002354* 001772 MOV #TQ,QPTR2
413 000502* 016700 002106 ;GET COUNT OF ACTIVE DEVICES
414 000506* 016002 002574* 1S: MOVB 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 DCRF BIT
417 000522* 052713 000100 BIC #4,4(R3) ;ENABLE RECEIVER INTERRUPTS
418 000526* 052763 000004 000004 INCB XBUP(R2),6(R3) ;ENABLE MAINT. MODE
419 000534* 105262 002414* 000006 MOVB XBUP(R2),6(R3) ;OUTPUT CHAR ONTO TX.
420 000540* 016263 002414* 000006 TXCNT ;UP COUNT OF CHARS OUTPUT.
421 000546* 105267 001704 INCB DCNT(R2) ;COUNT CHARACTERS OUTPUT ON THIS LINE
422 000552* 105262 002334* 000004 BIC #100,4(R3) ;ENABLE TX INTERRUPTS.
423 000556* 052763 000100 000004 DEC R0 ;COUNT QNE KICKED OFF.
424 000564* 100347 002334* 000004 BNE R0 ;COUNT ONE FOR NEXT ONE
425 000566* 100347 002334* 000004 MOV #6,CNTR ;INITIALIZE COUNTER TO WAIT AT LEAST
426 000570* 012767 000006 001664 10S: CLR R4 ;30 SECONDS BEFORE TIMING OUT
427 2S:
428 000576* 005004 10S:
429 000600* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.
430 000604* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
431 000608* 104407 000100 001641 CMFB #64,,TXCNT+1 ;64 TRANSMITTER INTERRUPTS?
432 000616* 003004 000100 001632 CMP #64,,RXCNT ;YES- BR R4 TO WAIT
433 000620* 022767 000100 001632 BLE 4S ;YES- 64 RECEIVER INTERRUPTS?
434 000626* 003405 000100 001632 BLE 4S ;YES- GO CHECK FOR ERRORS
435 000630* 003405 000100 001632 BNE 2S ;NO- WAIT SOME MORE
436 000632* 001362 000100 001622 DEC CNTR ;EACH PASS OF THE SMALL LOOP TAKES
437 000634* 005367 000100 001622 BNE 10S ;AT LEAST 5 SECONDS
438 000640* 001356 000100 001622 JMP ERRCHK ;BRANCH IF NOT DONE WITH 6 PASSES CF
439 4S:
440 000642* 000167 000272 4S:
441 ;TRANSMITTER INTERRUPT SERVICE - ENTERED VIA APPROPRIATE JSR TABLE
442 ;ENTRY WITH R5 POINTING TO THE CSR ADDRESS - CONTENTS OF R5
443 ;GETS QUEUED UP IN FIFO QUEUE AND ROUTINE RETURNS CONTROL BACK TO
444 ;THE MONITOR VIA A PIQO TO DEFER SERVICING XMTR AT LEVEL 0
445
446 000646* 010577 001614 TINT: MOV #R,QPTR1 ;STORE CONTENTS OF R5 IN THE QUEUE
447 000652* 062767 000002 001606 ADD #2,QPTR1 ;UPDATE THE QUEUE POINTER
448 000656* 022767 002414* 001600 CMP #TQ+40,QPTR1 ;POINTER AT END OF QUEUE?
449 000660* 001003 002354* 001570 MOV #R,QPTR1 ;PR IF NOT
450 000666* 001003 002354* 001570 1S: MOV #R,QPTR1 ;RESET THE POINTER
451 000670* 012763 002334* 001570 MOV (R6),R5 ;RESTORE THE OTHER GUY'S R5
452 ;-----
453 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 TSERV: MOV #R,QPTR2 ;GET POINTER FROM THE QUEUE
460 000706* 017700 001556 ADD #2,QPTR2 ;UPDATE THE QUEUE POINTER
461 000712* 062767 000002 001550 CMP #TQ+40,QPTR2 ;POINTER AT HIGH LIMIT
462 000720* 022767 002414* 001542 BNE 1S ;BR IF NOT
463 000730* 001003 002354* 001532 MOV #R,QPTR2 ;RESET THE POINTER
464 000736* 012001 000100 001532 MOV (R0),R1 ;MOV CSR ADDRESS INTO R1
465 000740* 011000 000100 001532 INCB TXCNT+1 ;COUNT TCTL TX INTERRUPTS.
466 000744* 012767 001511* 001532 INCB TCNT(R0) ;COUNT THE INTERRUPT
467 000748* 105267 002314* 001532 TST (R1) ;XMTR READY FLAG ASSERTED?
468 000754* 105711 000100 001472 BPL 4S ;BR IF NOT
469 000758* 122767 000100 001472 CMFB #64,,TXCNT ;64 CHARACTERS TRANSMITTED?
470 000764* 001427 000100 001472 BEQ 5S ;YES- BRANCH TO EXIT
471 000768* 001427 002414* 000002 MOVB XBUP(R0),2(R1) ;GENERATE NEXT DATA BYTE
472 000772* 116061 002414* 000002 INCB TXCNT ;LOAD THE XMTR BUFFER
473 000776* 105260 002334* 000002 INCB DCNT(R0) ;UP TOTAL COUNT OF CHARS OUTPUT.
474 000780* 010100 004000 000000 ;COUNT CHARACTERS OUTPUT ON THIS LINE
475 000784* 010167 177060 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
476 000788* 011167 177056 4S: MOV (R1),ACSR ;SAVE CSR ADDRESS
477 000792* 142711 000100 177056 BICB #100,(R1) ;SAVE CONTENTS OF THE CSR
478 000796* 012767 000011 177050 MOV #1,ERRTP ;DISABLE XMTR INTERRUPT
479 000800* 012767 000011 177050 ***** ;ILLGAL INTERRUPT
480 001036* 104405 000000* 000000 ***** ;XMTR FALSE INTERRUPT
481 001044* 104400 000000* 5S: EXIT$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
482 ;RECEIVER INTERRUPT SERVICE-ENTERED VIA APPROPRIATE JSR TABLE ENTRY
483 ;STORES PERTINENT INFORMATION IN THE RECEIVER STATUS TABLE THAT WILL
484 ;BE CHECKED IF 64 CHARACTERS HAVE BEEN RECEIVED
485 RINT: MOV R0,(R6) ;SAVE R0 AND PI ON THE STACK
486 001044* 104400 000000* 488 MOV R1,(R6)
489 001050* 010046 000100 001376 CMP #64,,RXCNT ;64 CHAR'S RECEIVED?
490 001054* 022767 000100 001376 BLE 4S ;YES- BRANCH
491 001058* 003414 000100 001376 MOV SVPTR,R0 ;NG- GET RCVR STATUS TABLE POINTER
492 001062* 003414 000100 001374 MOV (R5),R1 ;GET RCVR CSR ADDRESS
493 001066* 016700 000100 001374 MOV (R1),(R0)+ ;SAVE THE RCVR CONTENTS
494 001070* 011501 000002 001374 MOVB #R,(R0)+ ;SAVE THE RCVD DATA
495 001074* 116120 000002 001374 BIC (R1),-3(R0) ;REREAD CSR IN CASE ERROR SET BETWEEN THE
496 001100* 051160 177775 500 MOVB 2(R5),(R0)+ ;LAST TWO INSTRUCTIONS
497 001104* 116520 000002 502 MOV #R,SVPTR ;SAVE THE LINE NUMBER
498 001110* 010067 001350 503 MOV 2(R5),R5 ;SAVE THE UPDATED STATUS TABLE POINTER
499 001114* 016505 000002 504 INCB RCNT(R5) ;GET LINE NO. INTO R5
500 001118* 005267 001330 505 MOV (R5),R1 ;COUNT THE INTERRUPT FROM THIS LINE
501 001122* 012601 000002 506 MOV (R6)+,R1 ;INCREMENT RX INTERRUPT TOTAL COUNT.
502 001126* 012600 000002 507 MOV (R6)+,R1 ;RESTORE THE OTHER GUY'S REGISTER
503 001130* 006005 000002 508 RTI ;RETURN CONTROL BACK TO OTHER GUY
509 510 ;RECEIVER ERROR CHECKING AND CLEANUP ROUTINES
511
```



```
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623
```

001140 016700 001450
001144 016809 002574
001150 004767 000360
001154 042713 000100
001160 042763 000100 000004
001170 100365

THIS ROUTINE DISABLES INTERRUPTS FROM ALL ACTIVE LINES
ERRCHK: MOV ACTDEV,R0 ;GET COUNT OF ACTIVE DEVICES
1S: MOVB DEVTAB(R0),R2 ;GET ACTIVE LINE NO.
JSR PC,GETADR ;GC BUILD ADDRESS IN R3
BIC #100,(R3) ;TURN OFF RECEIVER
BIC #100,(R3) ;TURN OFF TRANSMITTER.
DEC R0 ;COUNT ONE GUY OFF
BPL 1S ;RR TIL ALL OFF

THIS ROUTINE SCANS THROUGH THE 64 ENTRY RECEIVER STATUS TABLE
CHECKING FOR AND REPORTING ANY ERRORS.
CHK1: CMP RXCNT,#64. ;MAKE SURE COUNT IS NO LARGER THAN TABLE
BLE 1S
MOV #64,RXCNT
2S: MOV RSTAB,R1 ;GET STATUS TABLE POINTER
CLR R0 ;INDICATE NO HARDWARE FAILURES YET.
MOVB 3(R1),R2 ;GET LINE NO. INTO R2
4S: TSTB (R1) ;POINT TO LD BYTE OF CSR
BMI 5S ;BE IF DCNR WAS SET
JSR PC,RCVERR ;SETUP FOR ERROR REPORT
MOV #1,ERRTYP ;ILLEGAL INTERRUPT

RDERS,BEGIN,NULL ;RECEIVER FALSE INTERRUPT

INCB RBUF(R2) ;BUMP EXPECTED DATA
TST R0 ;HARDWARE ERRORS?
BNE 6S
BICB #40,RBUF(R2) ;DO NOT REPORT DATA ERRORS THEN.
BICB #340,(R1) ;MASK OFF BITS <7:5> TO CHECK ONLY
CMPB RBUF(R2),2(R1) ;FIVE BITS ON
EQ 6S ;DID RCVD DATA CHECK OK?
JSR PC,BADT ;RR IF YES
6S: CMP (R1)+,(R1)+ ;GC REPORT DATA ERROR
DEC RXCNT ;GC INT R1 TO NEXT TABLE ENTRY
DEC R1 ;ALL CHARS RECEIVED CHECKED?
BNE 2S ;NC- GO CHECK NEXT ENTRY

THIS ROUTINE REPORTS ANY LINE RECEIVING AN INCORRECT NUMBER OF INTERRUPTS
CKLINS: MOV ACTDEV,R1 ;GET ACTIVE DEVICE COUNT
3S: MOVB DEVTAB(R1),R2 ;GET ACTIVE DEVICE LINE NO.
CMPB RCNT(R2),DCNT(R2) ;CORRECT NUMBER OF RCVR INTERRUPTS?
BEQ 4S ;RR IF YES
JSR PC,BADR ;GC REPORT BAD RCVR
CMPB TCNT(R2),DCNT(R2) ;CORRECT NUMBER OF XMTF INTERRUPTS?
BEQ 5S ;RR IF YES
JSR PC,BADT ;GC REPORT BAD XMTF
DEC R1 ;COUNT ONE GUY CHECKED
BPL 3S ;RR TIL ALL CHECKED
BR EXPS ;GC CHECK FOR END OF PASS

001316 016701 001272
001322 116102 002574
001326 126262 002274 002334
001334 001402
001342 126262 000064
001350 001402 002314 002334
001352 004767
001356 005301
001360 100360
001362 000530

RING: YRING
-1

MRING: .ASCIZ /RING SET- BAD LINE DROPPED/

```
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623
```

001404 042101 046040 047111
001412 020105 051104 050117
001420 001428 022504 000

.EVEN

ROUTINE TO REPORT BAD LINES (TOO MANY OR TOO FEW INTERRUPTS)
BADT: JSR PC,GETADR ;GC BUILD CSF ADDRESS
MOV R3,GETADR ;SAVE CSF ADDRESS
MOVB DCNT(R2),ACSR ;CHARACTERS XMTD
MOVB RCNT(R2),ASTAT ;# OF RCVR INTERRUPTS

001426 004767 000102
001432 010367 176442
001436 116267 002334 176436
001444 116267 002274 176432

001452 012767 000014 176426

001460 104405 000000 000000

MOV #14,ERRTYP

RDERS,BEGIN,NULL ;INCORRECT NUMBER OF RCVR INTERRUPTS

NOTE THAT CSRC VALUE IS # OF CHARACTERS
;TRANSMITTED, AND STATC VALUE IS # OF
;RECEIVER INTERRUPTS

001466 000207
RTS PC ;RETURN TO CALLER

001470 004767 000040
001474 022323 176376
001476 010367 176376
001502 116267 002334 176372
001510 116267 002314 176366

BADT: JSR PC,GETADR ;GC BUILD CSF ADDRESS
CMP (R3)+,(R3)+ ;MAKE IT A XMTF CSR ADDRESS
MOV R3,CSRA ;SAVE CSF ADDRESS
MOVB DCNT(R2),ACSR ;CHARACTERS XMTD
MOVB TCNT(R2),ASTAT ;# OF XMTF INTERRUPTS

001516 012767 000014 176362

001524 104405 000000 000000

MOV #14,ERRTYP

RDERS,BEGIN,NULL ;INCORRECT NUMBER OF XMTF INTERRUPTS

NOTE THAT CSRC VALUE IS # OF CHARACTERS
;TRANSMITTED, AND STATC VALUE IS # OF
;TRANSMITTER INTERRUPTS

001532 000207
RTS PC ;RETURN TO CALLER

001534 010203
001536 006303
001540 006303
001542 006303
001544 006703 176236
001550 000207

GETADR: MOV R2,R3 ;GET LINE NO.
ASL R3 ;BUILD CSP ADDRESS
ASL R3
ADD ADDR,R3
RTS PC ;RETURN TO CALLER

ROUTINE TO REPORT RCVR DATA COMPARE ERRORS
DATBAD: JSR PC,GETADR ;GC BUILD CSF ADDRESS
MOV R3,CSRA ;SAVE CSF ADDRESS
MOVB 2(R1),AWAS ;SAVE BAD DATA
TST (R1) ;GENERATE RCVR DATA ADDRESS
MOV R1,AWASADR ;SAVE ADDRESS OF BAD DATA
TST -(R1) ;RESET R1
MOV #R0,R5 ;GENERATE ADDRESS OF GOOD DATA
ADD R2,R5
MOVB (R5),ASB ;SAVE GOOD DATA
MOV R5,SEADR ;SAVE ADDRESS OF GOOD DATA

736	002724	000000		0		
737	002726	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	003010	004567	175632	JSR	R5,TINT	
757	003014	000000		0		
758	003016	000007		7		
759	003018	000000		0		
760	003024	000000		0		
761	003026	000010		10		
762	003028	004567	175612	JSR	R5,TINT	;LINK FOR LINE 10
763	003034	000000		0		
764	003036	000010		10		
765	003040	004567	176004	JSR	R5,RINT	;LINK FOR LINE 11
766	003042	000000		0		
767	003044	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	000010		10		
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	000010		10		
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	000446R	354#																		
SOPPAS	000046R	320#																		
SPOINT	000032R	314#																		
SPSIZ	000040R	1#	347																	
SR1	000016R	307#																		
SR2	000029R	308#																		
SR3	000022R	309#																		
SR4	000024R	310#																		
START	000248R	313#	356#																	
STAT	000026R	312#																		
STUP1	000440R	406#	646																	
SVPTR	002464R	410*	495	502*	667#															
SVRO	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	726	732	738	744	750	756	762	768	774	780						
TQ	002354R	411	412	450	452	462	464	655#	682											
TRPDFD	000022R	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

XKLAEO,XKLAEO/SQL/CRF:SYM=DDXCON,XKLAEO
 RUN-TIME: 1 2 3 SECONDS
 RUN-TIME RATIO: 30/4=6.2
 CORE USED: 7K (13 PAGES)