

-REM -

IDENTIFICATION

PRODUCT CODE: AC-E709I-MC
PRODUCT NAME: CXDLA10 DL11 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:

DLA IS AN IOMOD THAT EXERCISES UP TO SIXTEEN DL11 ASYNCHRONOUS INTERFACES. IT IS CAPABLE OF EXERCISING ALL DL11 MODELS. IT USES MAINTENANCE MODE TO TRANSMIT AND RECEIVE A BINARY COUNT PATTERN OUTPUT AND RECEIVED IN 64 CHARACTER BURSTS. THE MAJOR PORTION OF THE ERROR CHECKING IS DEFERRED TO PRIORITY LEVEL 0. ALL LINES SELECTED FOR TEST (UP TO 16 DL11'S WITH CONTIGUOUS ADDRESSES AND VECTORS) ARE ACTIVATED AND RUN CONCURRENTLY. ALL TRANSMIT AND RECEIVE ERRORS ARE REPORTED ON THE CONSOLE TTY.

NOTE:

THIS MODULE IS NOT DESIGNED TO EXERCISE ANY TYPE OF TERMINAL DEVICE. IT IS DESIGNED ONLY TO EXERCISE THE DL. ANY INFORMATION DISPLAYED OR PRINTED IS INCIDENTAL AND SHOULD BE IGNORED. IN MOST CASES THE TERMINAL DEVICE SHOULD BE PLACED IN THE OFF-LINE MODE.

2. REQUIREMENTS:

HARDWARE: AT LEAST ONE DL11 INTERFACE
NOTE: FOR THE DL11, THE TEST CONNECTOR MUST BE INSTALLED I

STORAGE:: DLA REQUIRES:
1. DECIMAL WORDS: 917
2. OCTAL WORDS: 1625
3. OCTAL BYTES: 3452

3. PASS DEFINITION:

ONE PASS OF THE DLA MODULE CONSISTS OF TRANSMITTING AND RECEIVING SOME CHARACTERS. THE EXACT NUMBER DEPENDS ON THE BAUD RATE

4. EXECUTION TIME:

VARIABLES WITH BAUD RATE BUT SHOULD TAKE ABOUT ONE
MINUTE TO COMPLETE ONE PASS WHEN RUNNING ALONE.
SPL MUST BE SET TO MATCH THE BAUD RATE OR THE
PASS TIMES WILL BE OFF.

5. CONFIGURATION PARAMETERS:

DEFAULT PARAMETERS:

DVA: 176500, VCT: 1, BR1: 4, BR2: 0, DVC: 1

REQUIRED PARAMETERS:

AT CONFIGURATION TIME THE USER MUST SPECIFY:

VCT: VECTOR ADDRESS OF FIRST DL11
DVC: NO OF DL11'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^M THIS SECTION OF CODE SETS UP THE VECTORS OF ALL SELECTED LINES TO POINT TO THE APPROPRIATE JSR INSTRUCTION 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. STOP: THIS ROUTINE INITIALIZES ALL TABLE 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 ERCHK.

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 SERVICE VICING IS DONE LATER, WHERE THE SERVICE CODE IS EXECUTED AT PRIORITY 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 RE-
CEIVER STARTUP TABLE TESTS FOR THE
END OF A 64 CHAR BURST SEQUENCE AND
THEN EXECUTES AN "RTI". IT ALSO COUNTS
RECEIVE INTERRUPTS IN A SEPARATE
COUNTER FOR EACH LINE.
SEPARATE COUNTER.

G. FRRCHK:

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. ENVARLES
FOR ALL ACTIVE LINES
2. SCAN THROUGH THE RECEIVER STATUS TABLE
(64 ENTRIES OF TWO WORDS EACH) TO CHECK
FOR AND REPORT:

- 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. EMPS:

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. RSTAR:

THIS IS A 128 WORD STATUS TABLE CON-
SISTING 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. APPROPRIATE BYTE GETS INCREMENTED DURING DEFERRED INTR. SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.

D. DLNT: 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,PRINT(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. THE USER CAN USE THE "MOD" COMMAND TO DUMP THE TABLES

DLA10.P11 12-OCT-78 12:01

BUFFERS DESCRIBED IN 7.2 TO OBTAIN MORE DETAILED ERROR INFORMATION.

- B. THE USER CAN MODIFY (DLA 14) "DVID1" TO SELECT OR DESELECT INDIVIDUAL DL1'S.
- C. SRI IS USED TO SPECIFY THE BAUD RATE AND CHARACTER SIZE WITH WHICH THE DLA MODULE IS TO WORK. THE DEFAULT IS 9600 BAUD WITH AN 8-BIT CHARACTER SIZE.

SRI	BIT1	BIT0	CHAR SIZE
0	0	0	8-BIT
0	0	1	7-BIT
1	0	0	6-BIT
1	1	1	5-BIT

- 2. BAUD RATE: BITS 2-14 ARE USED. ONE AT MOST SHOULD BE SET; IF MORE THAN ONE BIT IS SET, THE RIGHT-MOST WILL BE USED.

BAUD	BIT SET
9600	NONE
7200	2
4800	3
2400	4
1800	5
1200	6
600	7
300	8
150	9
75	10
37.5	11
18.75	12
9.375	13
4.6875	14

- 3. EXAMPLES:

SRI=5 MEANS 7-BIT CHARACTER AT 7200 BAUD
SRI=12 MEANS 6-BIT CHARACTER AT 4800 BAUD
SRI=1000 MEANS 8-BIT CHARACTER AT 200 BAUD
SRI=0 (DEFAULT) MEANS 8-BIT CHARACTER AT 9600 BAUD

DLAI DEC/X11 SYSTEM EXERCISER MODULE
XDLAI0.P11 12-OCT-78 12:01

NACY11 30A(1052) 12-OCT-78 16:27 PAGE 9

SEQ 0008

9. NON-STANDARD PRINTOUTS:

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

```
DL11 A-D DEC/X11 EXERCISER MODULE
000000 IOMOD <DLAI > 176500,1,4,0,26
000000 MODULF 140000,DLAI,176500,1,4,0,26
; TITLE DLAI DEC/X11 SYSTEM EXERCISER MODULE
; DDACOM VERSION 6 LIST BIN
*****
000000 BEGIN: -ASCII /DLAI / ;MODULE NAME.
000000 046104 044501 040 XFLAG: -BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000000 000000 ADDR: 176500+0 ;1ST DEVICE ADDR.
000019 000001 VECTOR: -10 ;1ST DEVICE VECTOR.
000017 000200 BR1: -BYTE PRTY4+0 ;1ST BR LEVEL.
000013 000000 BR2: -BYTE PRTY+0 ;2ND BR LEVEL.
000014 000001 DVID1: +1 ;DEVICE INDICATOR 1.
000016 000000 SR1: OPEN ;SWITCH REGISTER 1
000020 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: MODDSP ;MODULE STACK POINTER.
000034 000000 PASCNT: 0 ;PASS COUNTER.
000036 000000 ICOUNT: 0 ;# OF ITERATIONS PER PASS=0
000040 000000 ICMOUT: 0 ;LOC TO COUNT ITERATIONS
000042 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046 000000 SIFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052 000000 SVSCNT: 0 ;# OF SVS ERRORS ACCUMULATED
000054 000000 RANNUM: 0 ;HOLDS RANMOD # WHEN RAND MACRO IS CALLED
000056 000000 CONFIG: 0 ;RESERVED FOR MONITOR USE
000060 000000 RES1: 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.
000070 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 CSR.
000102 000000 SADR: OPEN ;ADDR OF GOOD DATA, OR
000104 000000 WASADR: OPEN ;ADDR OF BAD DATA, OR
000106 000000 ASSTAT: OPEN ;STATUS REG CONTENTS.
000108 000000 ERHTYP: 0 ;TYPE OF ERROR
000110 000000 ASB: OPEN ;EXPECTED DATA.
000112 000522 AWAS: OPEN ;ACTUAL DATA.
000114 000000 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
```

```
000116 000000 WDRP: OPEN ;WORDS FROM MEMORY PER ITERATION
000120 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000122 000026 IDNUM: 26 ;MODULE IDENTIFICATION NUMBER=26
;REPT SPSIZ ;MODULE STACK STARTS HERE.
;NLIST
;WORD 0
;LIST
;FNDR

000224 MODDSP:
*****
THIS ROUTINE SETS UP THE VECTORS FOR ALL SELECTED LINES TO POINT
TO THE APPROPRIATE JSR IN THE JSR LINK TABLE AND SETS
THE ITERATION COUNT TO MATCH BAUD RATE SPEC. IN SR1
START: MOV #64,WDTO ;64 (10) WORDS PER ITERATION TO MEM
MOV #64,WDRP ;64 (10) WORDS FROM MEM PER ITERATION
MOV #128,INTR ;64 INTERRUPTS PER ITERATION
MOV SR1,R1 ;GET SR1 VALUE
BIC #17774,R1 ;MASK OUT SIZE PART
BEQ 8S ;DEFAULT SETTING
MOV #200,R0 ;SET UP MASK FOR 7-BIT
DEC R1 ;COUNT BITS MASKED
BEQ 8S ;EXIT WHEN DONE
ASR R0 ;SHIFT AND REPLICATE HIGH BIT
BR 7S ;AGAIN?
MOV R0,SIZE ;SET SIZE MASK
CLR R1 ;INIT POINTER
MOV SR1,R1 ;GET SR1
ASR R1 ;SHIFT SR1 TO
ASR R1 ;GET BAUD RATE
19S ;9600 BAUD SET
ADD #2,R0 ;ELSE RAMP POINTER
ROR R1 ;LOOK FOR SR1 BIT
BCS 19S ;LEAVE IF WE FOUND IT
BR 19S ;ELSE DO IT AGAIN
MOV CNTBL(R0),ICOMT ;SET UP ITER COUNT
17S: MOV VECTOR,R0 ;SET R0 TO POINT TO THE 1ST VECTOR
MOV DVID1,R1 ;LOAD R1 WITH DEVICE SELECTION PARAMETER
MOV R1,DEVICE ;INITIALIZE TO RECORD ANY LINES DROPPED
MOV #JSRTAB,R2 ;SET UP R2 TO POINT TO JSR TABLE
15S: ASR R1 ;SHIFT SELECT BIT INTO "C" TABLE
BCC 3S ;RR IF NOT SELECTED
MOV R2,(R0)+ ;SET UP RCVR PTRN POINTER
MOV BR1,(R0)+ ;SET UP RCVR PRIORITY LEVEL
TSTR (R0)+ ;MOVE POINTER
ADD #10,R2 ;POINT R2 TO XMTN ENTRY IN JSR TABLE
43S: MOV R2,(R0)+ ;SET UP XMTN PTRN POINTER
MOV BR1,(R0)+ ;SET UP XMTN PRIORITY LEVEL
TSTR (R0)+ ;MOVE POINTER
44S: ADD #10,R2 ;POINT R2 TO RCVR ENTRY FOR NEXT LINE
45S: CMP #JSRTAB+400,R2 ;IS THE POINTER AT THE END OF THE TABLE?
46S: BR 15S ;RR IF NOT
47S: BR SETCSR ;GO SET UP CSR ADDRESSES
48S: ADD #10,R0 ;UPDATE VECTOR POINTER
49S: ADD #20,R2 ;UPDATE JSR TABLE POINTER
449: BR 25S ;GO CHECK FOR END OF TABLE
```

```

450 ;THIS ROUTINE SETS UP THE JSR TABLE SUCH THAT THE APPROPRIATE
451 ;CSR ADDRESS IS INCLUDED AS THE 3RD WORD OF EACH ENTRY
452 000432- 016700 177350 SETCSR: MOV ADDR,R0 ;GET THE FIRST CSR ADDRESS INTO R0
453 000436- 016701 177352 MOV DIVD1,R1 ;LOAD R1 WITH THE DEVICE SELECTION PARAMETER
454 000442- 001002 ENDS,BEGIN ;RR IF DVC OK
455 000444- 104410 000000- 1S: MOV #JSRTAB+4,R2 ;POINT R2 TO CSR ADDRESS ENTRY
456 000450- 016702 003056- 2S: ASR R1 ;SHIFT SELECT BIT INTO "C"
457 000454- 002201 BCC R1 ;RR IF LINE NOT SELECTED
458 000456- 103014 MOV R0,(R2) ;RR IF LINE NOT SELECTED
459 000460- 010012 CMP (R0)+,(R0)+ ;GENERATE XMTX CSR ADDRESS IN R0
460 000464- 022020 ADD #10,R2 ;POINT TO XMTX SLOT IN JSR TABLE
461 000466- 022020 MOV R0,(R2) ;PUT XMTX CSR ADDRESS IN THE TABLE
462 000470- 010012 CMP (R0)+,(R0)+ ;GENERATE NEXT RCVR CSR ADDRESS IN R0
463 000474- 022020 ADD #10,R2 ;POINT TO RCVR SLOT IN JSR TABLE
464 000476- 062702 000010 3S: CMP #JSRTAB+404,R2 ;IS POINTER BEYOND END OF TABLE?
465 000500- 022702 003456- BNE RST ;RR IF NOT
466 000504- 001163 BR RST ;RR SET UP ACTIVE DEVICE TABLE.
467 000506- 000410 4S: ADD #10,R0 ;UPDATE CSR ADDRESS
468 000510- 062700 000010 ADD #20,R2 ;UPDATE JSR TABLE POINTER
469 000520- 006787 BR 3S ;GO TEST FOR END OF TABLE
470
471 ;THIS ROUTINE CLEARS BUFFERS AND TABLES, INITIALIZES FLAGS, AND STARTS
472 ;UP ALL SELECTED LINES.
473
474 000522- 005767 177306 RESTR: TST PASCNT ;HAVE WE BEEN THRU START?
475 000526- 001636 BEQ START ;RR THERE IF WE HAVEN'T
476 000530- 001636 JSR PC,CLRBUF ;GO CLEAR XMTX AND RCVR. BUFFERS
477 000534- 004767 002170 STUPL: JSR PC,CLRTAB ;GO CLEAR THE ACTIVE DEVICE TABLE.
478 000540- 004767 002176 JSR PC,CLRTAB ;GO CLEAR TABLES AND QUEUES
479 000544- 005064 002140 CLR TXCNT ;CLEAR TX TOTAL INTERRUPT COUNTER.
480 000548- 005064 002140 CLR RXCNT ;CLEAR RX TOTAL INTERRUPT COUNTER.
481 000552- 012767 002134 481 000556- 012767 002604- 002134 MOV #4,R3 ;INITIALIZE RCVR STATUS TABLE POINTER
482 000560- 012767 002604- 002124 482 000564- 012767 002604- 002124 MOV #4,R3 ;SET UP XMTX FIFO QUEUE POINTERS
483 000568- 116002 003026- 1S: MOV #10,R2 ;GET COUNT OF ACTIVE DEVICES
484 000572- 116002 003026- 1S: MOV #10,R2 ;GET IN ACTIVE LINE NUMBER
485 000576- 004767 000112 JSR PC,GETADR ;GO BUILD CSR ADDRESS IN R3
486 000580- 004767 000112 TST 2(R3) ;READ RCVR DRR TO FLUSH DONE BIT
487 000584- 004767 000112 BLS #100,(R3) ;ENABLE RECEIVER INTERRUPTS
488 000588- 004767 000112 INCR XBUF(R2) ;ENABLE MAINT. MODE
489 000592- 004767 000112 MOVX XBUF(R2),6(R3) ;OUTPUT CHAR ONTO TX.
490 000596- 004767 000112 INCR TXCNT ;UP COUNT OF CHARS OUTPUT.
491 000600- 004767 000112 INCR RXCNT ;COUNT CHARACTERS OUTPUT ON THIS LINE
492 000604- 004767 000112 BLS #100,(R3) ;ENABLE TX INTERRUPTS.
493 000608- 004767 000112 DEC R0 ;COUNT ONE KICKED OFF.
494 000612- 004767 000112 BPL 1S ;GO TEST FOR NEXT ONE
495 000616- 004767 000112 MOV #6,CNTR ;INITIALIZE COUNTER TO WAIT AT LEAST
496 000620- 004767 000112 CLR R4 ;30 SECONDS BEFORE TIMING OUT
497 000624- 004767 000112 10S: CLR R4
498 000628- 004767 000112 2S:
499
500 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
501 000632- 004767 000112 BREAKS,BEGIN ;IF CONTINUING AT NEXT INSTRUCTION.
502 000636- 004767 000112 CMPR #64, TXCNT+1 ;64 TRANSMITTER INTERRUPTS?
503 000640- 004767 000112 8GT 3S ;NO- BRANCH TO WAIT
504 000644- 004767 000112
505 000648- 004767 000112

```

```

506 000714- 022767 000100 001770 CMP #64,RXCNT ;YES- 64 RECEIVER INTERRUPTS?
507 000722- 003405 BLE 4S ;YES- GO CHECK FOR ERRORS
508 000726- 003405 DEC R4 ;TIMEOUT?
509 000730- 003405 BNE 2S ;NO- WAIT SOME MORE
510 000734- 001356 001760 DEC CNTR ;EACH PASS OF THE SMALL LOOP TAKES
511 ;AT LEAST 5 SECONDS
512 000738- 001356 BNE 10S ;BRANCH IF NOT DONE WITH 6 PASSES OF
513 ;THE SMALL COUNTER
514 000736- 000167 000306 4S: JMP ERRCHK
515
516 ;TRANSMITTER INTERRUPT SERVICE - ENTERED VIA APPROPRIATE JSR TABLE
517 ;ENTRY WITH R5 POINTING TO THE CSR ADDRESS. - CONTENTS OF R5
518 ;GETS QUEUED UP IN FIFO QUEUE AND ROUTINE RETURNS CONTROL BACK TO
519 ;THE MONITOR VIA A PING TO DEPR
520 TINT: MOV R5,QPTR1 ;STORE CONTENTS OF R5 IN THE QUEUE
521 000742- 010577 001752 ADD #2,QPTR1 ;UPDATE THE QUEUE POINTER
522 000746- 022767 002944- 001736 CMP #10,QPTR1 ;POINTER AT END OF QUEUE?
523 000750- 001003 BNE #10,QPTR1 ;RR IF NOT
524 000754- 022767 002944- 001736 MOV #10,QPTR1 ;RESET THE POINTER
525 000758- 012605 001726 MOV (R0)+,R5 ;RESTORE THE OTHER GUY'S R5
526 000762- 012605 001726 PIRQS,BEGIN,TSERV ;QUEUE UP TO CONTINUE AT TSERV AND RTI
527
528
529
530
531 ;DEFERRED XMTX SERVICE - THIS ROUTINE RETRIEVES POINTER TO CSR ADDRESS
532 ;FROM THE FIFO QUEUE AND SERVICES THE LINE AT LEVEL 0
533 TSERV: MOV #QPTR2,R0 ;GET POINTER FROM THE QUEUE
534 001002- 017700 001714 ADD #2,QPTR2 ;UPDATE THE QUEUE POINTER
535 001006- 022767 002644- 001706 CMP #10,QPTR2 ;POINTER AT HIGH LIMIT
536 001010- 001003 BNE 1S ;RR IF NOT
537 001014- 012767 002604- 001670 MOV #10,QPTR2 ;RESET THE POINTER
538 001018- 011000 MOV (R0)+,R1 ;MOV CSR ADDRESS INTO R1
539 001022- 012767 001647 MOV R0,(R0) ;MOV LINE # INTO R0
540 001026- 022767 002544- INCR TXCNT+1 ;COUNT TOTAL TX INTERRUPTS.
541 001030- 011000 TST TXCNT ;COUNT THE INTERRUPT
542 001034- 011000 BPL 4S ;XMTX READY FLAG ASSERTED?
543 001038- 010011 BR 4S ;RR IF NOT
544 001042- 122767 000100 001630 CMPR #64, TXCNT ;64 CHARACTERS TRANSMITTED?
545 001046- 001427 BEQ 5S ;YES- BRANCH TO EXIT
546 001050- 105260 002644- 000002 INCR XBUF(R0) ;GENERATE NEXT DATA BYTE
547 001054- 105260 001610 MOVX XBUF(R0),2(R1) ;LOAD THE XMTX BUFFER
548 001058- 105260 002564- INCR DCNT(R0) ;UP TOTAL COUNT OF CHARS OUTPUT.
549 001062- 104400 000000- 5S: EXITS,BEGIN ;COUNT CHARACTERS OUTPUT ON THIS LINE
550 001066- 010197 176762 MOV R1,CSRA ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
551 001070- 011167 176762 MOV (R1),RCSR ;SAVE CSR ADDRESS
552 001074- 142711 000100 BICB #100,(R1) ;SAVE CONTENTS OF THE CSR
553 001078- 012767 000011 176754 MOV #11,CRRTYP ;DISABLE XMTX INTERRUPT
554 ;*****
555 001132- 104405 000000- 000000 HDRSRC,BEGIN,NULL ;*****
556 ;*****
557 001140- 104400 000000- 5S: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
558
559 ;RECEIVER INTERRUPT SERVICE-ENTERED VIA APPROPRIATE JSR TABLE ENTRY
560 ;STORES PERTINENT INFORMATION IN THE RECEIVER STATUS TABLE THAT WILL
561

```



```

DLAI0.P11 12-OCT-78 12:01
674
675 001620 051045 047111 020107
676 001626 042523 026524 041040
677 001634 042101 046040 047111
678 001642 020105 051104 050117
679 001650 042520 022504 000
680 001656
681
682
683 001656 004767 000102
684 001652 010367 176112
685 001666 116267 002564 176206
686 001674 116267 002564 176202
687
688 001702 012767 000014 176176
689
690 001710 104405 000000 000000
691
692
693
694
695
696 001716 000207
697
698 001720 004767 000040
699 001724 022323 176146
700 001726 010367 002564 176142
701 001732 116267 002564 176136
702 001740 116267 002564 176132
703
704 001746 012767 000014 176132
705
706 001754 104405 000000 000000
707
708
709
710
711
712 001762 000207
713
714 001764 010203
715 001766 006303
716 001770 010367 006303
717 001772 006303
718 001774 006703 176006
719 002000 000207
720
721
722 002002 004767 177756
723 002005 010367 176065
724 002009 010367 000002 176070
725 002020 005721
726 002022 010167 176056
727 002026 005741
728 002030 012705 002664
729 002034 060205

```

```

MRING: .ASCIZ /%RING SET- BAD LINE DROPPED%/
.EVEN
;ROUTINE TO REPORT BAD LINES (TOO MANY OR TOO FEW INTERRUPTS)
BADR: JSR PC,GETADR ;GO BUILD CSR ADDRESS
;R1,CSRA ;SAVE CSR ADDRESS
MOV R2,DCNT(R2),ACSR ;CHARACTERS XMTD
MOV R3,RCNT(R2),ASTAT ;# OF RCVR INTERRUPTS
MOV #14,ERRTYP ;WRONG # OF INTERRUPTS
;*****
;RDERS,BEGIN,NULL ;INCORRECT NUMBER OF RCVR INTERRUPTS
;*****
;NOTE THAT CSRC VALUE IS # OF CHARACTERS
;TRANSMITTED, AND STATC VALUE IS # OF
;RECEIVER INTERRUPTS
RTS PC ;RETURN TO CALLER
BADT: JSR PC,GETADR ;GO BUILD CSR ADDRESS
;R1,CSRA ;SAVE CSR ADDRESS
CMB (R3),R3+ ;MAKE IT A XMTD CSR ADDRESS
MOV R2,DCNT(R2),ACSR ;CHARACTERS XMTD
MOV R3,RCNT(R2),ASTAT ;# OF XMTD INTERRUPTS
MOV #14,ERRTYP ;WRONG # OF INTERRUPTS
;*****
;RDERS,BEGIN,NULL ;INCORRECT NUMBER OF XMTD INTERRUPTS
;*****
;NOTE THAT CSRC VALUE IS # OF CHARACTERS
;TRANSMITTED, AND STATC VALUE IS # OF
;TRANSMITTER INTERRUPTS
RTS PC ;RETURN TO CALLER
GETADR: MOV R2,R3 ;GET LINE NO.
ASL R3 ;BUILD CSR ADDRESS
ASL R3
ADD ADDR,R3
RTS PC ;RETURN TO CALLER
;ROUTINE TO REPORT RCVR DATA COMPARE ERRORS
DATBAD: JSR PC,GETADR ;GO BUILD CSR ADDRESS
;R1,CSRA ;SAVE CSR ADDRESS
MOV R2,R1,AWAS ;SAVE RCVR DATA ADDRESS
TST (R1),AWAS ;GENERATE RCVR DATA ADDRESS
MOV R1,AWASADR ;SAVE ADDRESS OF BAD DATA
TST (R1) ;RESET R1
MOV #R0DE,R5 ;GENERATE ADDRESS OF GOOD DATA
ADD R2,R5

```

```

DLAI0.P11 12-OCT-78 12:01
730 002036 111567 176044
731 002042 010567 176034
732
733 002046 104404 000000
734
735 002052 000207
736
737
738 002054 005200
739 002056 004767 177702
740 002062 010367 176012
741 002066 011167 176010
742 002072 000207
743
744
745 002074
746 002074 104413 000000
747
748
749
750 002100 012700 002664
751 002104 012701 002644
752 002110 112021
753 002112 022700 002704
754 002116 001374
755 002120 000167 176410
756
757
758
759 002124 000200
760 002524 000010
761 002544 000010
762 002564 000010
763
764 002604 000020
765
766 002644 000010
767 002664 000010
768
769
770
771 002704 000000
772 002706 000000
773 002710 000000
774
775 002712 000000
776 002714 000000
777 002716 000000
778 002720 000000
779 002722 000000
780
781
782
783 002724 012700 002644
784 002730 005020
785 002732 022700 002704

```

```

MOV R5,ASB ;SAVE GOOD DATA
MOV R5,BADR ;SAVE ADDRESS OF GOOD DATA
;*****
;RDERS,BEGIN ;DATA ERROR!!!
;*****
RTS PC ;RETURN TO CALLER
;ROUTINE TO SETUP FOR RECEIVER ERROR PRINTOUTS
RCVFR: INC R0 ;INDICATE HARDWARE ERROR-
;R1,CSRA ;GO BUILD CSR ADDRESS
MOV R2,DCNT(R2),ACSR ;STUFF IT IN CSRA
MOV R3,RCNT(R2),ACSR ;GET CONTENTS IN ACSR
RTS PC ;RETURN TO CALLER
;THIS ROUTINE CHECKS FOR AND REPORTS END OF PASS
ENPS: ENDS,BEGIN ;SIGNAL END OF ITERATION,
;MONITOR SHALL TEST END OF PASS
;THIS ROUTINE RESTARTS EACH 64 CHAR XFR SEQUENCE
RESYNC: MOV #RBUF,R0 ;RESYNC DATA FOR NEXT PASS
MOV #R0DE,R1
IS: MOV (R0),R1+
CMP #RBUF+20,R0 ;DONE 16 BYTES?
BNE IS ;RR IF NOT
JMP STUP1 ;RESUME.
;TABLES AND BUFFERS
RSTAB: .BLKW 128. ;128 WORD(64 ENTRIES)RCVR STATUS TABLE
RCNT: .BLKW 8. ;RCVR INTERRUPT COUNTERS
TCNT: .BLKW 8. ;XMTD INTERRUPT COUNTERS
DCNT: .BLKW 8. ;CHARACTER COUNTERS
TQ: .BLKW 16. ;16 WORD XMTD FIFO QUEUE
XRBUF: .BLKW 8. ;16 BYTE XMTD DATA BUFFERS
RBUF: .BLKW 8. ;16 BYTE RCVR DATA BUFFERS
;POINTERS, CONSTANTS, AND VARIABLES
SIZE: OPEN ;LOW BYTE IS MASK FOR VARIABLE 5-8 BIT COMPARE
COUNT: OPEN ;END OF PASS COUNTER
TXCNT: OPEN ;TX TOTAL INTERRUPTS COUNTER (HIGH BYTE).
;TOTAL CHARACTERS TRANSMITTED (LOW BYTE)
RXCNT: OPEN ;RX TOTAL INTERRUPTS COUNTER.
CVR: OPEN ;BREAK LOOP COUNTER
SVPTR: OPEN ;TEMP STORAGE FOR RSTAB POINTER
QPTR1: OPEN ;XMTD FIFO QUEUE POINTER - LOAD
QPTR2: OPEN ;XMTD FIFO QUEUE POINTER - UNLOAD
;SUBROUTINE TO CLEAR DATA BUFFERS AT BEGINING OF EACH NEW PASS
CLRBUF: MOV #RBUF,R0 ;SET UP R0 TO POINT TO BEGINING
IS: CMB (R0),R0 ;CLEAR A WORD
CMP #RBUF+20,R0 ;END OF RCVR BUFFER?

```

```

786 002736* 001374      BNE 1S          ;RR TIL ALL CLEAR
787 002740* 000207      RTS PC          ;RETURN TO CALLER
788
789
790 002742* 012700 002124* ;SUBROUTINE TO CLEAR TABLES AND QUEUES
791 002746* 005020      CLRAR: MOV #RSTAR,R0 ;SET UP R0 TO POINT TO BEGINING
792 002750* 022700 002644* 1S: CLR (R0)+      ;CLEAR A WORD
793 002754* 001374      CMP #TQ+40,R0    ;END?
794 002756* 000207      BNE 1S          ;RR IF NOT
795                      RTS PC          ;RETURN TO CALLER
796
797
798
799 002760* 005000      ;THIS ROUTINE SETS UP AN ACTIVE DEVICE TABLE TO REMEMBER HOW MANY
800 002762* 005100      ;AND WHICH LINES WERE ACTIVE DURING TEST - IT IS USED DURING THE
801 002764* 005001      ;ERROR CHECKING ROUTINES FOR VARIOUS PURPOSES
802 002766* 005101      DTAB: CLR R0          ;SET UP R0 AS TOTAL LINE COUNTER
803 002770* 016702 000054  CLR R0          ;INITIALLY SET TO MINUS ONE
804 002774* 005200      CDR R1          ;SET UP R1 AS ACTIVE LINE COUNTER
805 002776* 000020 1S:  CDR R1          ;INITIALLY SET TO MINUS ONE
806 003002* 001003      MOV DVICE,R2     ;GET DEVICE SELECTION PARAMETER
807 003004* 010167 000036  INC R0          ;COUNT ONE DEVICE
808 003010* 000207      CMP #20,R0      ;16 LINES CHECKED?
809 003012* 006202      BNE 2S          ;RR IF NOT
810 003014* 103367      MOV R1,ACTDEV    ;SAVE THE COUNT OF ACTIVE LINES
811 003016* 005201 2S:  RTS PC          ;RETURN TO CALLER
812 003020* 110061      ASR R0          ;SHIFT SELECT BIT INTO "C"
813 003024* 000763      INC R1          ;RR IF NOT SELECTED
814                      INC R1          ;COUNT ACTIVE LINE
815                      MOV#R R0,DEVTAB(R1) ;STORE ACTIVE LINE NO. IN THE TABLE
816                      BR 1S          ;GO TEST NEXT LINE
817
818
819
820
821 003052* 004567 176066  DEVTAB: .RLX# 8.      ;16 BYTE ACTIVE DEVICE TABLE
822 003056* 000000 176066  ACTDEV: OPEN      ;STORES COUNT OF NO. OF ACTIVE LINES MINUS ONE
823 003060* 000000      DVICE: OPEN      ;DEVICE SELECTION INDICATOR
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841

```

```

842 003142* 004567 175574  JSR R5,TINT
843 003146* 000000      0
844 003150* 000003      3
845 003154* 000000 175766  JSR R5,RINT      ;LINK FOR LINE 4
846 003156* 000000      0
847 003160* 000004      4
848 003162* 004567 175554  JSR R5,TINT
849 003166* 000000      0
850 003170* 000004      4
851 003172* 004567 175746  JSR R5,RINT      ;LINK FOR LINE 5
852 003176* 000000      0
853 003200* 000905      5
854 003202* 000905 175534  JSR R5,TINT
855 003206* 000000      0
856 003210* 000005      5
857 003212* 004567 175726  JSR R5,RINT      ;LINK FOR LINE 6
858 003216* 000000      0
859 003220* 000006      6
860 003222* 004567 175514  JSR R5,TINT
861 003226* 000000      0
862 003230* 000006      6
863 003232* 004567 175706  JSR R5,RINT      ;LINK FOR LINE 7
864 003236* 000000      0
865 003240* 000007      7
866 003242* 004567 175474  JSR R5,TINT
867 003246* 000000      0
868 003250* 000007      7
869 003252* 004567 175666  JSR R5,RINT      ;LINK FOR LINE 10
870 003256* 000000      0
871 003260* 000010      10
872 003262* 004567 175454  JSR R5,TINT
873 003266* 000000      0
874 003270* 000010      10
875 003272* 004567 175646  JSR R5,RINT      ;LINK FOR LINE 11
876 003276* 000000      0
877 003300* 000011      11
878 003302* 004567 175434  JSR R5,TINT
879 003306* 000000      0
880 003310* 000011      11
881 003312* 004567 175626  JSR R5,RINT      ;LINK FOR LINE 12
882 003316* 000000      0
883 003320* 000012      12
884 003322* 004567 175414  JSR R5,TINT
885 003326* 000000      0
886 003330* 000012      12
887 003332* 004567 175606  JSR R5,RINT      ;LINK FOR LINE 13
888 003336* 000000      0
889 003340* 000013      13
890 003342* 004567 175374  JSR R5,TINT
891 003346* 000000      0
892 003350* 000013      13
893 003352* 004567 175566  JSR R5,RINT      ;LINK FOR LINE 14
894 003356* 000000      0
895 003360* 000014      14
896 003362* 004567 175354  JSR R5,TINT
897 003366* 000000      0

```


GTPPAS= 104415	404#																		
GWBPPS= 104414	404#																		
HRDCNT= 000044R	369#																		
HRDRRS= 104405	404#																		
HRDPAS= 000050R	401#	555	609	627	690	706													
ICONT= 000036R	366#	428*																	
ICONT= 000040R	367#																		
IDNM= 000124R	396#																		
INIT= 000030R	363#																		
INTR= 000120R	395#	409*																	
JSRTAR= 003052R	432#	443	456	465	821#														
MAP22S= 104416	404#																		
MODNAM= 000000R	350#																		
MNDSP= 000224R	364#	402#																	
MRING= 001620R	672#	675#																	
MSGNS= 104403	404#	613																	
MSGSS= 104402	404#																		
MSGSS= 104401	404#																		
NULL= 000000	404#	555	609	627	690	706													
OPEN= 000000	351#	357	358	359	360	377													
	386	390	391	393	394	395	378	379	380	381	382	383	384						
	777	778	779	816	817														
DTOAS= 104420	404#																		
PASCHT= 000034R	355#	475																	
PIRGS= 000004	404#	527																	
POPPSP= 005726	404#																		
POPPSP2= 022626	404#																		
PRTV= 000000R	355#	404#																	
PRTV0= 000000	404#																		
PRTV1= 0000040	404#																		
PRTV2= 000100	404#																		
PRTV3= 000140	404#																		
PRTV4= 000200	354#	404#																	
PRTV5= 000240	404#																		
PRTV6= 000300	404#																		
PRTV7= 000340	404#																		
PS= 177776	404#																		
PSW= 177776	404#																		
PUSH2= 005746	404#																		
QPTR1= 002720R	483#	520*	521*	522	524*	778#													
QPTR2= 002722R	484#	532	533*	534	536*	779#													
HANDS= 104414R	404#																		
RANUM= 00414R	377#																		
RRUP= 002664R	629#	632*	634	728	750	753	767#	785											
RCNT= 002524R	578#	644	686	760#															
RCVRRP= 002054R	606#	624	738#																
RESR= 00054R	46	477#																	
RESRRT= 000522R	392	475#																	
RESVHC= 002100R	750#																		
RESI= 000056R	375#																		
RESI= 000032R	364#																		
RING= 001614R	613	672#																	
RINT= 001144R	563#	821	827	833	839	845	851	857	863	869	875	881	887						
	893	899	905	911															
RSTAB= 002124R	482	601	759#	790															

RSTRT= 000112R	392#																		
RXCNT= 002712R	481#	506	565	579*	598	600*	638*	775#											
SRADR= 000102P	385#	731*																	
SPTCR= 000432R	445#	452#																	
SIZE= 00704R	418#	632	633	771#															
SDFCNT= 000042R	368#																		
SDFPRS= 104406	404#																		
SDFPAS= 000046R	370#																		
SPHINT= 000032R	364#																		
SPSTZ= 000040	397																		
SR1= 000016R	357#	410	420																
SR2= 000020R	358#																		
SR3= 000022R	359#																		
SR4= 000024R	360#																		
START= 000224R	363#	407#	476																
STAT= 000026R	362#																		
STUPL= 000534R	478#	755	567	576*	777#														
SVPTR= 002716R	482#																		
SVRO= 000062R	377#																		
SVR1= 000064R	378#																		
SVR2= 000066R	379#																		
SVR3= 000070R	380#																		
SVR4= 000072R	381#																		
SVR5= 000074R	382#																		
SVR6= 000076R	383#																		
SVSCNT= 000052R	372#																		
TCNT= 002544R	540#	647	702	761#															
TINT= 000742R	520#	824	830	836	842	848	854	860	866	872	878	884	890						
	846	902	908	914															
	483	484	522	524	534	536	764#	792											
TJ= 002604R	404#																		
TRPDFD= 000022	527	532#																	
TSERV= 001002R	480#	493#																	
TXCNT= 002710R	353#	429	504	539*	543	547*	773#												
VECTOR= 000010R	353#																		
WASADR= 000104R	387#	726*																	
WDFR= 000116R	394#	408*																	
WDTJ= 000114R	393#	407*																	