

.REM\_

IDENTIFICATION

PRODUCT CODE: AC-E718I-MC  
PRODUCT NAME: CXDUAI0 DU11 DEC/X11 MODULE  
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

DUA IS AN IOMOD THAT EXERCISES UP TO EIGHT 8-BIT SYNCHRONOUS LINE INTERFACES (DU11) BY TRANSMITTING A STANDARD BINARY COUNT PATTERN USING THE MAINTENANCE MODE FEATURE. THE RECEIVED DATA IS COMPARED WITH THE TRANSMITTED DATA AND ANY ERRORS ARE REPORTED VIA THE CONSOLE TTY. ALL AVAILABLE INTERFACES (UP TO 8) ARE ACTIVATED AND RUNNING SIMULTANEOUSLY.

2. REQUIREMENTS

HARDWARE: DU11 SYNCHRONOUS INTERFACE  
STORAGE: DUA REQUIRES:  
1. DECIMAL WORDS: 823  
2. OCTAL WORDS: 1467  
3. OCTAL BYTES: 3156

3. PASS DEFINITION

ONE PASS OF THE DUA MODULE CONSISTS OF TRANSMITTING AND RECEIVING 40000 (8) 8-BIT CHARACTERS (TOTAL)

4. EXECUTION TIME

DUA RUNNING ALONE ON A PDP11/05 PROCESSOR TAKES APPROXIMATELY 1 MINUTES TO COMPLETE ONE PASS.

5. CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DEVADR: 1, VECTOR:1, BR1:5, BR2:5, DEVCNT:1

REQUIRED PARAMETERS: SR1 = 1 = ASYNCRONOUS (ISOCRONOUS)  
SR1 = 0 = SYNCRONOUS

DVA= DEVICE ADDRESS OF THE FIRST DU11

VCT= VECTOR ADDRESS OF THE FIRST DU11

6. DEVICE/OPTION SETUP

NONE: NO DEVICE IS REQUIRED IN MAINTENANCE MODE

7. MODULE OPERATION

TEST SEQUENCE:

- A. TEST UP TO 8 POSSIBLE DEVICES FOR SELECTION
- B. STORE THE NO. OF DEVICES TO BE TESTED AND SET UP THE VECTORS FOR THESE DEVICES
- C. TURN ON RECEIVER INTERRUPT ENABLE, TRANSMITTER INTERRUPT ENABLE, AND MAINTENANCE MODE FOR ALL ACTIVE DEVICES.
- D. INITIAL TRANSMITTER INTERRUPT SERVICE:
  - 1.) TEST FOR FALSE INTERRUPT (READY (0)); REPORT ERRORS
  - 2.) OUTPUT NEXT CHARACTER TO EACH ACTIVE DEVICE
  - 3.) RETURN TO MONITOR TO WAIT FOR RECEIVER INTERRUPT.
- E. RECEIVER INTERRUPT SERVICE:
  - 1.) TEST FOR FALSE INTERRUPT (DONE (0)); REPORT ERRORS
  - 2.) COMPARE INPUT/OUTPUT DATA; REPORT ERRORS
  - 3.) RETURN TO MONITOR TO WAIT FOR TRANSMITTER INTERRUPT
- F. REPEAT D AND E UNTIL 40000.(TOTAL) CHARACTERS HAVE BEEN PROCESSED
- G. AT END OF PASS TURN OFF ALL ACTIVE DEVICES AND RESTART AT B

8. OPERATION OPTIONS

A. LOCATION DVID1 (DUA 14) MAY BE CHANGED TO SELECT ANY COM-  
BINATION OF DEVICES BIT0=DEVO, BIT1=DEV1 .....BIT

;BIT DE IF DVID1 IS INITIALLY 0 DUA WILL BE DROPPED FROM TEST.

B. LOCATIONS START+2 AND PASS+14(8) MAY BE MODIFIED TO INCREASE  
OR DECREASE THE TOTAL NUMBER OF CHARACTERS PROCESSED PER PASS

9. NON STANDARD PRINTOUTS

NONE: ALL PRINTOUTS HAVE STANDARD FORMATS AS DESCRIBED IN THE  
DEC/X11 DOCUMENT.

```
-  
;BIT DEFINITIONS  
BIT5=10000  
BIT4=4000  
BIT13=2000  
BIT12=10000  
BIT11=4000  
BIT10=200  
BIT9=1000  
BIT8=400  
BIT7=200  
BIT6=100  
BIT5=40  
BIT4=20  
BIT3=10  
BIT2=4  
BIT1=2  
BIT0=1  
  
;PROCESSER LEVELS  
LEVEL7=340  
LEVEL6=300  
LEVEL5=240  
LEVEL4=200  
LEVEL3=140  
LEVEL2=100  
LEVEL1=40  
LEVEL0=000  
  
;REGISTER DEFINITIONS  
;REGISTER EQUATES  
RXCSR=0  
PARCSR=2  
RXDBUF=2  
TXCSR=4  
TXDBUF=6  
  
;RXCSR BIT DEFINITIONS  
DSC=BIT14 ;DATA SET CHANGE  
RING=BIT14 ;RING  
CTS=BIT13 ;CLR TO SEND  
CARDET=BIT12 ;CARRIER DETECT  
REACT=BIT11 ;REC ACTIVE  
SRD=BIT10 ;SEC REC DATA  
DSR=BIT9 ;DATA SET RDY  
STPSYN=BIT8 ;STRIP SYNC  
RXDONE=BIT7 ;REC DONE  
RXINTE=BIT6 ;REC INTR ENABLE  
DSINTE=BIT5 ;DSC INTR ENABLE  
SYNSCH=BIT4 ;SYNC SEARCH  
SID=BIT3 ;SEC XMIT DATA  
RTS=BIT2 ;REQ TO SEND  
DTR=BIT1 ;DATA TERM RDY  
VOID=BIT0  
  
;RXDBUF BIT DEFINITIONS  
RXERR=BIT15 ;REC ERROR
```

```
OVRUN=BIT14 ;OVERRUN  
FRMERR=BIT13 ;FRAME ERROR  
PARERR=BIT12 ;PARITY ERROR  
  
;PARCSR BIT DEFINITIONS  
PAREN=BIT9 ;PARITY ENABLE  
EVPAR=BIT8 ;EVEN PARITY SENSE  
  
;PARCSR WRD DEFINITIONS  
SYNINT=30000 ;SYNC EXTERNAL MODE  
SYNXT=20000 ;SYNC INTERNAL MODE  
ISYMOD=0 ;ISOC MODE  
FIVE=0 ;WORD LENGTH 5 BITS  
SIX=2000 ;WORD LENGTH 6 BITS  
SEVEN=4000 ;WORD LENGTH 7 BITS  
EIGHT=6000 ;WORD LENGTH 8 BITS  
NOPAR=0 ;NO PARITY  
DDPAR=1000 ;ODD PARITY  
EVEPAR=1400 ;EVEN PARITY  
  
;TXCSR BIT DEFINITIONS  
DNA=BIT15 ;DATA NOT AVAILABLE  
MTDATA=BIT14 ;MAINT DATA  
CLK=BIT13 ;CLK  
BITW=BIT10 ;BIT WINDOW  
MRESET=BIT8 ;MASTER RESET  
TXDONE=BIT7 ;XMIT DONE  
TXINTE=BIT6 ;XMIT INTR ENABLE  
DNAINTR=BIT5 ;DNA INTR ENAB  
SEND=BIT4 ;SEND  
HDXEN=BIT3 ;HDX/FDX  
BREAK=BIT0 ;BREAK  
  
;TXCSR WRD DEFINITIONS  
USER=0 ;USER MODE  
MINT=4000 ;MAINT INT MODE  
MEXT=10000 ;MAINT EXT MODE  
SYSTST=14000 ;SYSTEM TEST MODE
```

```

;LIST SEQ BIN
;SET UP VECTOR (RETURN ADDRESS(PC)) PC = INTR SERV. AREA.
IOMOD <DUAI> 5,5,2000,32
MODULE 140000 DUAI 5,5,2000,32
TITLE DUAI DEC/X11 SYSTEM EXERCISER MODULE
DDACOM VERSION 6 23-MAY-78
;*****LIST BIN*****
BEGIN:
MODNAM: .ASCII /DUAI / ;MODULE NAME.
XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WRUFF USAGE
ADDR: 1+0 ;1ST DEVICE ADDR.
VECTOR: 1+0 ;1ST DEVICE VECTOR.
BRI: .BYTE PRTV5+0 ;1ST BR LEVEL.
BR2: .BYTE PRTV5+0 ;2ND BR LEVEL.
DVID: +1 ;DEVICE INDICATOR 1.
SR1: OPEN ;SWITCH REGISTER 1.
SR2: OPEN ;SWITCH REGISTER 2.
SR3: OPEN ;SWITCH REGISTER 3.
SR4: OPEN ;SWITCH REGISTER 4.
;*****LIST BIN*****
STAT: 140000 ;STATUS WORD.
INIT: START ;MODULE START ADDR.
SPCNT: 0 ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER.
ICOUNT: 2000 ;# OF ITERATIONS PER PASS=2000
ICOUNT: 0 ;LOC TO COUNT ITERATIONS
SOFCHT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
SRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
RANNUM: 0 ;# OF SVS FRMS ACCUMULATED
CONFIG: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;RESERVED FOR MONITOR USE
SVRO: OPEN ;LOC TO SAVE R0.
SVR1: OPEN ;LOC TO SAVE R1.
SVR2: OPEN ;LOC TO SAVE R2.
SVR3: OPEN ;LOC TO SAVE R3.
SVR4: OPEN ;LOC TO SAVE R4.
SVR5: OPEN ;LOC TO SAVE R5.
SVR6: OPEN ;LOC TO SAVE R6.
CSRA: OPEN ;LOC OF CURRENT CSR.
SBADR: 0 ;ADDR OF GOOD DATA, OR
ACSR: OPEN ;CONTENTS OF CSR.
WASADR: 0 ;ADDR OF BAD DATA, OR
ASTAT: 0 ;STATUS OF CONTENTS.
ERRTYP: 0 ;TYPE OF ERROR.
ASB: OPEN ;EXPECTED DATA.
AMA: OPEN ;ACTUAL DATA.
RESTR: RESTR ;START ADDRESS AFTER END OF PASS
WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION

```

```

IDNUM: 32 ;MODULE IDENTIFICATION NUMBER=32
;MODULE STACK STARTS HERE.
.REPT SPSIZ
.NLIST
0
.LIST
.ENDR
MODSP:
;*****LIST BIN*****
START: MOV #8,WDTO ;8 WORDS TO MEM/ITERATION
MOV #3,WDFR ;8 WORDS FROM MEM/ITERATION
MOV #3,INTR ;3 INTERRUPTS/ITERATION
MOV #1,SELECT ;COPY THE DEVICE SELECTION PARAMETER
BNE RESTR ;ARE ANY SELECTED? IF YES, START PROCESSING
;IF NO,DROP THE MODULE
;
; ONLY START W.D. TIMER ONCE
; RESET THE COUNT OF TRANSMITTER INTERRUPTS
; RESET THE RECEIVER COUNT
; CLR FLAG INDICATING NO. OF DU'S DONE
; NO DU'S SELECTED
; SET UP Q POINTERS
; " " "
; COPY ACTIVE SELECTION PARAMETER
; R1 = VECTOR ADDRESS
; R2 = LINK: USR TABLE WITH OFFSET
; ISOLATE A SELECTION FLAG IN THE CARRY BIT
; IF SELECTED GO SET UP VECTORS
; IF NO MORE, CONTINUE PROCESSING
; IF MORE, UPDATE POINTERS FOR NEXT DEVICE
; GO PROCESS NEXT DU11
; SET UP VECTOR RETURN ADDRESS(RCV)
; SET UP VECTOR PRIORITY (RCV)
; INCR. POINTER
; UPDATE NEW LINK ADDRESS
; SET UP VECTOR RETURN ADDRESS (XMT)
; SET UP VECTOR PRIORITY (XMT)
; INCR. POINTER
; UPDATE NEW LINK ADDRESS FOR DUXMT
; GO CHECK FOR MORE DEVICES
;
; THIS CODE WILL CLEAR ALL OF THE WRITE BUFFER AREA
SETUP1: MOV #103,R0 ;COUNT REQUIRED TO GO THRU
;ALL DATA STORAGE BUFFERS
MOV #DULIN,R3 ;STARTING ADDRESS OF
;DATA BUFFER LOCATIONS.
1S: CLR (R3)+ ;CLEAR DATA RUFF REG
DEC R0 ;ARE THERE MORE TO CLEAR?
BNE 1S ;NO GO BACK & DO THE REST
;
; THIS CODE WILL SELECT WHICH LINES (<18>) HAVE
; BEEN SELECTED FOR TEST & TRANSMIT SYNC TO START
; TESTING ALL LINES.

```

```
339  
340 000434* 012700 000010 INITIAL:MOV #10,R0 ;SET COUNT VALUE  
341 000440* 016701 177342 MOV ADDR,R1 ;R1=THE ADDRESS OF FIRST DUAL  
342 000444* 012702 002404* MOV #DVAL1,R2 ;R2=ADDRESS OF TABLE OF ADDRESSES  
343 000450* 012703 002364* MOV #DVAL2,R3 ;R3=ADDRESS OF TABLE OF BUFFER ADDRESSES  
344 000454* 012704 002304* MOV #DVAL3,R4 ;POINT R4 TO THE FIRST BUFFER  
345 000460* 010122 000000 1$:MOV R1,(R2)+ ;DVALX=DEVICE ADDR. CODE  
346 000462* 010423 000000 MOV R4,(R3)+ ;BUFF POINTER HAS START OF LINE  
347 ;BUFF STORAGE  
348 000464* 062701 000010 ADD #10,R1 ;UPDATE  
349 000470* 062704 000020 ADD #20,R4 ;UPDATE  
350 000474* 005300 000000 DEC R0 ;CNT DOWN  
351 000476* 001370 000000 BNE 1$ ;NOT DONE GO BACK FOR MORE  
352  
353  
354 000500* 012767 000010 002410 START2:MOV #8,NODVTS ;SET UP COMPLEMENT OF NUMBER OF LINES TESTED  
355 000506* 016701 177274 MOV ADDR,R1 ;GET DEVICE ADDRESS  
356 000512* 012700 002425* MOV #LNSVN1+1,R0 ;SET UP R0 TO POINT TO LNSVN LOC.  
357 000516* 012703 002444* MOV #LNCNT1,R3  
358 000522* 016702 002366 MOV SELECT,R2 ;COPY ACTIVE SELECTION FLAGS  
359 000526* 006202 000000 BCS R2 ;TEST IS THIS DEVICE ON LINE  
360 000530* 103406 000000 ASR 4$ ;YES, GO SETUP MORE  
361 000532* 001414 000000 BEQ ACTIVATE ;GO START ACTIVE DEVICES  
362 000534* 005113 000002 1$:COH (R3,2) ;SET LINCNT FLAG IF DEVICE NOT SELECTED  
363 000542* 005723 000000 ADD #2,R0 ;UPDATE THE TABLE POINTERS  
364 000544* 000770 000000 TST (R3)+  
365 000546* 112710 000004 4$:BR DS ;TRY THE NEXT DEVICE  
366 000552* 012713 010420 MOV #4,(R0) ;LOAD SYNC COUNT IN LNSVN X(HIGH BYTE)  
367 ;COUNT=17 HIGH FOR TRANSMIT (16 CHAR. PLUS 1)  
368 000556* 005367 002334 DEC NODVTS ;LOW FOR RECEIVE BIT 3=TRANS. W/INTERRUPT  
369 000562* 000765 000000 BR 1$ ;KEEP TRACK OF NUMBER OF UNTESTED DEVICES  
370 000564* 016702 002324 ACTIVATE:MOV SELECT,R2 ;TRY NEXT DEVICE  
371 000570* 006202 000000 3$:ASR R2 ;GET ACTIVE DEVICE FLAGS  
372 000572* 103404 000000 BCS 5$ ;ISOLATE A SELECTION FLAG  
373 000574* 001421 000010 6$:BEQ TMRSET ;IF SET, GO START DEVICE  
374 000576* 001094 000000 ADD #10,R1 ;IF ALL SET GO START TIMER  
375 000602* 000772 000000 BR 1$ ;NO, ITS ISOCRONOUS  
376 000604* 012761 014020 5$:MOV #SYSTST1SEND,TXCSR(R1) ;GO START NEXT DEVICE  
377 000612* 005767 177200 TST SR1 ;ARE YOU RUNNING SYNCHRONOUSLY?  
378 000616* 001094 000000 BR 1$ ;NO, ITS ISOCRONOUS  
379 000620* 012761 036026 000002 MOV #NYNINTEIGHTINOPAR126,PARCSR(R1) ;SET SYNC INTERNAL,  
380 ;EIGHT BITS PER CHAR,NO PARITY,26=SYNC  
381 000626* 000403 000000 BR 2$ ;BRANCH AROUND  
382 000630* 012761 006026 000002 1$:MOV #ISYMODIEIGHTINOPAR126,PARCSR(R1) ;SET ISOCRONOUS MODE,  
383 ;EIGHT BITS PER CHAR,NO PARITY,26= SYNC  
384 000636* 012761 000520 000000 2$:MOV #RINTEN1STPSYNISVNSCH,RXCSR(P1) ;SET REC INTERRUPT ENABLE,  
385 ;STRIP SYNC SEARCH SYNC  
386 000644* 052761 000100 000004 BIS #XINTE,TXCSR(R1) ;ENABLE TRANSMITTER INTERRUPT  
387 000652* 000751 000000 BR 6$  
388  
389  
390  
391  
392 ;THIS IS THE WATCHDOG TIMER.  
393 ;IF ALL GOES WELL THE TIMEDOUT FLAG WILL BE CLEARED.IF IT IS NOT IN THE PRESCRIPED TIME,  
394 ;A MESSAGE WILL BE SENT .
```

```
395  
396 000654* 005767 002274 TMRSET: TST TFLAG ;HAVE WE BEEN HERE BEFORE?  
397 000660* 001002 000000* BNE 2$ ;BR IF NOT  
398 000662* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.  
399 000666* 005067 002262 2$:CLR TFLAG ;DON'T COME THRU HERE AGAIN  
400 000672* 012767 002246 MOV #1,TMOTFLG ;SET THE TIMEDOUT FLAG  
401 000700* 012767 000005 002242 MOV #5,TRCNT ;SET ENOUGH TIME FOR INTERRUPTS  
402 000706* 005067 002240 CLR TICKER ;START TIMEOUT  
403  
404 000712* 104407 000000* 1$:BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....  
405 000716* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
406 000722* 005767 002220 TST TMOTFLG ;ALL DONE?  
407 000726* 001410 000000 BEQ FINISH ;IF YES, GO FINISH UP  
408 000730* 005367 002216 DEC TICKER ;IF NO, TICK SOME TIME  
409 000734* 001366 000000* BNE 1$ ;BREAK AGAIN,IF NO TIMEOUT  
410 000736* 104403 000000* MSGNS,BEGIN,HUNG ;ASCII MESSAGE CALL WITH COMMON HEADER  
411 000744* 104410 000000* ENDS,BEGIN ;  
412 000750* 000000* FINISH: ENDS,BEGIN ;SIGNAL END OF ITERATION.  
413 000750* 104413 000000* ENDITS,BEGIN ;MONITOR SHALL TEST END OF PASS  
414  
415 000754* 000167 177302 JMP RSTRRT  
416  
417 ;THIS CODE WILL ANSWER THE XMT INTERRUPT REQUEST  
418 ; FOR SERVICE  
419  
420  
421 000760* 011577 001760 DUXMT: MOV (R5),QAD1 ;STORE CONTENTS OF R5 IN THE QUEUE  
422 000764* 001416 000002 ADD #2,QAD1 ;UPDATE THE QUEUE POINTER  
423 000772* 022767 002744* 001744 CMP #10+40,QAD1 ;POINTER AT END OF QUEUE?  
424 001000* 001003 000000 BNE 1$ ;BR IF NOT  
425 001002* 012767 002704* 001734 1$:MOV #10,QAD1 ;RESET THE POINTER  
426 001010* 012605 000000 MOV (R6),R5 ;RESTORE THE OTHER GUY'S R5  
427  
428 001012* 000004 000000* 001020* ;IRQS,BEGIN,QOUT ;QUEUE UP TO CONTINUE AT QOUT AND RTI  
429  
430  
431  
432 ;DEFERRED XMT SERVICE - THIS ROUTINE RETRIEVES POINTER TO CSR ADDRESS  
433 ;FROM THE FIFO QUEUE AND SERVICES THE LINE AT LEVEL 0  
434  
435  
436  
437 001020* 017700 001722 QOUT: MOV @QAD2,R0 ;GET POINTER FROM THE QUEUE  
438 001024* 062767 000002 ADD #2,QAD2 ;UPDATE THE QUEUE POINTER  
439 001032* 022767 002744* 001706 CMP #10+40,QAD2 ;POINTER AT HIGH LIMIT  
440 001040* 001003 000000 BNE 1$ ;BR IF NOT  
441 001042* 012767 002704* 001676 1$:MOV #10,QAD2 ;RESET THE POINTER  
442 001050* 016001 002404* MOV #DVAL1,(R0),R1 ;R1 = R0 WITH OFFSET VALUE  
443 001054* 032761 004000 000000 BIT #RECACT,RXCSR(R1) ;TEST TO GIVE MORE SYNC CHARS  
444 001062* 001426 000000 BEQ DUXMT2 ;BRANCH IF IN SYNC  
445 001064* 105360 002445* DECB LNCNT1+1(R0) ;SENT 16 CHARACTERS?  
446 001070* 001416 000000 BEQ DUXMT3 ;ES-BRANCH  
447 001072* 005267 002036 INC TRCNT ;INCR TRANSMIT COUNT  
448 001076* 116061 002424* MOV LNSVN1(R0),TXDBUF(R1) ;SENT DATA  
449 001104* 105260 002424* INCB LNSVN1(R0) ;FORM NEXT DATA WORD  
450 001110* 122760 000026 002424* CMPB #26,LNSVN1(R0) ;IS IT SYNC CHARACTER?
```

```
451 001116 001006 BNE DUXMT4 ;INC-BRANCH
452 001120 005260 INCB LNSYN1(R0) ;INC DATA WORD AGAIN
453 001124 000403 BR DUXMT4 ;BRANCH TO EXIT
454 001126 042761 000100 000004 DUXMT3: BIC #100, TXCSR(R1) ;CLR INT ENB
455 001134 001000 DUXMT4: EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
456 001140 104400 000000 000006 DUXMT2: TSYNC, TXDBUF(R1) ;XMT SYNC CHAR (TSYNC)
457 001140 116761 001746 000006 DECB LNSYN1+1(R0) ;DEC SYNC COUNTER
458 001146 105360 002425 000000 XMTRTN: XMTRTN ;EXIT IF SYNC COUNT NOT ZERO
459 001154 001003 000004 002425 BNE MOVB #4, LNSYN1+1(R0) ;RE-INITIALIZE SYNC COUNTER
460 001162 112760 000000 000000 XMTRTN: EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
461 001162 104400 000000 000000 ; REQUEST FOR SERVICE
462 001166 010246 DURCV: MOV R2, -(SP) ;SAVE REG 2 ON STACK
463 001170 010346 MOV R3, -(SP) ;SAVE REG. 3 ON STACK
464 001172 010446 MOV R4, -(SP) ;SAVE REG. 4 ON STACK
465 001174 011503 MOV #R5, R3 ;GET OFFSET
466 001176 016304 002404 MOV DVAD1(R3), R4 ;R3 = R4 DEVICE CODE OFFSET VALUE
467 001204 105714 TSTB (R4) ;IS DONE SET
468 001206 104425 BHI DURCV1 ;DONE SET -SERV DONE REQUEST
469 001206 010467 176666 MOV R4, CSRA ;SHOW CSR ADDR.
470 001212 011467 176664 MOV (R4), ACSR ;CONTENTS OF CSR
471 001216 005064 000000 CLR RXCSR(R4) ;
472 001224 012603 MOV (SP)+, R3 ;RESTORE STACK
473 001226 012602 MOV (SP)+, R2 ;
474 001230 012605 MOV (SP)+, R5 ;
475 001232 000004 000000 001240 PIRQS, BEGIN, FIRER ; QUEUE UP TO CONTINUE AT FIRER AND RTI
476 001240 012767 000011 176640 FIRER: MOV #11, ERRTYP
477 001246 104405 000000 000000 HRDRS, BEGIN, NULL ;FALSE INTERRUPT
478 001254 000167 000662 JMP DROP ;TURN OFF THIS DEVICE
479 001260 016467 000002 001636 DURCV1: MOV RXDBUF(R4), RCVDAT ;SAVE RXDRUF
480 001266 005767 001632 TST RCVDAT ;ARE THERE ANY ERRORS ?
481 001272 100145 BPL READ ;NO
482 001274 105363 002424 INCB LNSYN1(R3) ;UPDATE XMT DATA
483 001300 105263 002445 INCB LNCNT1+1(R3) ; " " ACTIVE COUNT
484 001304 010467 176570 MOV R4, CSRA ;CSR ADDR.
485 001310 016767 001610 MOV (R4), ACSR ;CONTENTS CSR
486 001316 005064 000000 CLR RXCSR(R4) ;
487 001322 012604 MOV (SP)+, R4 ;RESTORE STACK
488 001324 012603 MOV (SP)+, R3 ;
489 001326 012602 MOV (SP)+, R2 ;
490 001330 012605 MOV (SP)+, R5 ;
491 001332 032767 040000 001564 BIT #OVERRN, RCVDAT ;TEST FOR OVERRUN
492 001340 001403 BEQ IS ;NO
493 001342 000004 000000 001412 PIRQS, BEGIN, OVERR ; QUEUE UP TO CONTINUE AT OVERR AND RTI
494 001350 032767 020000 001546 1$: BIT #FRMERR, RCVDAT ;TEST FOR FRAME ERROR
```

```
507 001356 001403 BEQ 2$ ;NO
508 001360 000004 000000 001456 PIRQS, BEGIN, FRERR ; QUEUE UP TO CONTINUE AT FRERR AND RTI
509 001366 032767 010000 001530 2$: BIT #PARERR, RCVDAT ;TEST FOR PARITY ERROR
510 001374 001403 BEQ 3$ ;NO
511 001376 000004 000000 001522 PIRQS, BEGIN, PAERR ; QUEUE UP TO CONTINUE AT PAERR AND RTI
512 001404 3$:
513 001404 000004 000000 001566 PIRQS, BEGIN, EBIT15 ; QUEUE UP TO CONTINUE AT EBIT15 AND RTI
514 001412 012767 000021 176466 OVERR: MOV #21, ERRTYP
515 001420 104405 000000 000000 HRDRS, BEGIN, NULL ;OVERRUN ERROR
516 001426 005267 001506 INCB OVERRU ;COUNT ERRORS
517 001432 022767 000003 001500 CMP #3, OVERRU ;3 ERRORS ?
518 001440 001403 BEQ IS ;YES
519 001446 005067 176622 JMP RESTR ;NO RETRY MODULE
520 001452 000167 000464 JMP DROP ;GO Deselect THIS MODULE
521 001456 012767 000022 176422 FRERR: MOV #22, ERRTYP
522 001464 104406 000000 000000 HRDRS, BEGIN, NULL ;FRAME ERROR
523 001472 005267 001444 INCB FRAMER ;COUNT ERRORS
524 001476 022767 000003 001436 CMP #3, FRAMER ;3 ERRORS ?
525 001504 001403 BEQ IS ;YES
526 001506 000167 176556 JMP RESTR ;NO RETRY MODULE
527 001512 005067 001424 1$: CLR FRAMER ;CLEAN UP
528 001516 000167 000420 JMP DROP ;GO Deselect THIS MODULE
529 001522 012767 000023 176356 PAERR: MOV #23, ERRTYP
530 001530 104406 000000 000000 HRDRS, BEGIN, NULL ;PARITY ERROR
531 001536 005267 001402 INCB PARTER ;COUNT ERRORS
532 001542 022767 000003 001374 CMP #3, PARTER ;3 ERRORS ?
533 001550 001403 BEQ IS ;YES
534 001552 000167 176512 JMP RESTR ;NO RETRY MODULE
535 001556 005067 001362 1$: CLR PARTER ;CLEAN UP
536 001562 000167 000354 JMP DROP ;GO Deselect THIS MODULE
537 001566 012767 000000 176312 EBIT15: MOV #0, ERRTYP
538 001574 104405 000000 000000 HRDRS, BEGIN, NULL ;RECEIVE FLAG ERROR
539 001602 000167 000334 JMP DROP ;DESELECT THIS MODULE
540 001606 032764 004000 000000 READ: BIT #REACT, RXCSR(R4) ;IS DEVICE ACTIVE
```



```

563 001614* 001447 BEQ BADDONE ;GET OUT DEVICE NOT READY
564 001616* 005002 CLR R2 ;CLEAR BYTE PTRER
565 001620* 066302 002464* ADD VRFLG1(R3),R2 ;GET BYTE OFFSET
566 001624* 066302 002364* ADD DULIN(R3),R2 ;ADD=DATA BUFF ADDR
567 001630* 116412 000002 MOVB 2(R4),(R2) ;DATA => DATA BUFF
568 001634* 122712 000026 CMPB #26,(R2) ;SKP IF SYNC BIT
569 001640* 001430 BEQ RCVRTN ;
570 001642* 001270 INC RCVRTN ;
571 001646* 105263 002464* INC VRFLG1(R3) ;
572 001652* 105363 002444* DECB LNCNT1(R5) ;
573 ;
574 ;
575 001656* 001021 BNE RCVRTN ;CHECK HAVE WE TRANSFERRED ALL
576 ; DATA WORDS
577 001660* 005064 000004 CLR TXCSR(R4) ;THIS LINE NOT DONE RECEIVING
578 001664* 005064 000000 CLR RXCSR(R4) ;ALL DATA TRANSFERS
579 001674* 026767 001230 001230 CMP NO,DU,DUDN ;MASTER RESET (DEVICE)
580 001702* 001007 BNE RCVRTN ;COUNT THIS DU
581 001704* 012604 MOV (SP)+,R4 ;ARE WE DONE CHCK'ING
582 001706* 012603 MOV (SP)+,R3 ;RESTORE STACK
583 001710* 012602 MOV (SP)+,R2 ;
584 001712* 012605 MOV (SP)+,R5 ;
585 -----
586 001714* 000004 000000* 002014* FIRQS,BEGIN,CHCK ; QUEUE UP TO CONTINUE AT CHCK AND RTI
587 ;
588 ;
589 ;
590 ;
591 ;
592 ;
593 ;
594 ;
595 ;
596 ;
597 ;
598 ;
599 ;
600 001722* 012604 RCVRTN: MOV (SP)+,R4 ;RESTORE STACK POINTER
601 001724* 012603 MOV (SP)+,R3 ;
602 001726* 012602 MOV (SP)+,R2 ;
603 001732* 000002 RTI (SP)+,R5 ;
604 ;
605 ;
606 ;
607 ;
608 ;
609 ;
610 ;
611 ;
612 ;
613 ;
614 001734* 010467 176140 176134 BADDONE: MOV MOV R4,CSRA ;SAVE CSR ADDRESS
615 001746* 005064 000004 MOV RXCSR(R4),ACSR ;CONTENTS OF CSR
616 001752* 005064 000000 CLR TXCSR(R4) ;COLLAPSE THIS DEVICE IT A'INT WORKIN'
617 001756* 012604 CLR RXCSR(R4) ;
618 001762* 012602 MOV (SP)+,R4 ;RESTORE STACK POINTER
619 001764* 012602 MOV (SP)+,R3 ;
620 001766* 012605 MOV (SP)+,R2 ;
621 001768* 012605 MOV (SP)+,R5 ;
622 -----
623 001766* 000004 000000* 001774* FIRQS,BEGIN,BDONE ; QUEUE UP TO CONTINUE AT BDONE AND RTI
624 ;
625 ;
626 ;
627 ;
628 ;
629 ;
630 ;
631 ;
632 ;
633 ;
634 ;
635 ;
636 ;
637 ;
638 ;
639 ;
640 ;
641 ;
642 ;
643 ;
644 ;
645 ;
646 ;
647 ;
648 ;
649 ;
650 ;
651 ;
652 ;
653 ;
654 ;
655 ;
656 ;
657 ;
658 ;
659 ;
660 ;
661 ;
662 ;
663 ;
664 ;
665 ;
666 ;
667 ;
668 ;
669 ;
670 ;
671 ;
672 ;
673 ;
674 ;

```

```

619 002014* 005001 CHCK: CLR R1
620 002016* 005002 CLR R2
621 002020* 005000 CLR R0 ;CLEAR R0:R0 WILL BE
622 ; USED AS OFFSET
623 002022* 012767 000020 001072 CHCK1: MOV #20,COUNT ;FOR COUNTING NO OF
624 ; CHAR. READ
625 ; CLR BUFF POINTER
626 002030* 005002 CLR R2
627 002032* 012701 002444* MOV #LNCNT1,R1
628 002036* 105711 1S: TSTB (R1)
629 002040* 001402 BEQ CHCK2
630 002044* 000774 CMP (R1)+,(R2)+
631 002046* 010200 BR 1S
632 002050* 012607 002364* CHCK2: MOV R2,R0 ;R0 WILL HOLD LINE NO./2
633 002054* 112607 001046 MOV DULIN(R2),R2 ;R2=START ADDR. THIS LINE BUFF
634 002060* 126722 001042 MOV (R2),CHECKR ;GET FIRST CHAR.
635 002064* 001410 CONTNU: CMPB CHECKR,(R2)+ ;CHECK DATA & INCR. POINTER
636 002066* 001093 BEQ 1S ;THIS WORD GOOD GO CHECK MORE
637 002070* 001093 BNE ERRRT ;WAS IT SYNC CHAR.
638 002076* 005267 001024 INC CHECKR ;NO GO REPORT ERROR
639 002102* 005302 DEC R2 ;YES UPDATE CHECKR
640 002104* 005267 BR CONTNU ;UPDATE DULIN BUFFER POINTER
641 002106* 005267 1S: INC CHECKR ;GO BACK & CHECK REAL DATA
642 002112* 005367 001014 DEC COUNT ;SET UP FOR NEXT BYTE TEST
643 002116* 001360 BNE CONTNU ;ONE MORE BYTE HAS BEEN TESTED
644 002120* 005267 000772 INC NODVTS ;NOT DONE YET GO CHECK MORE
645 ; THIS LINE DONE ADD 1 TO
646 ; NO. OF DEVICES TESTED
647 002130* 022767 000010 000760 MOV #100777,(R1)
648 002136* 001506 CMP #10,NODVTS ;HAVE ALL LINES BEEN TESTED
649 002140* 000730 BR CHCK1 ;GO TO END PASS CODING
650 ;
651 ;
652 ;
653 ;
654 002142* 104403 000000* 002162* DROP: MSGNS,BEGIN,1S ;ASCII MESSAGE CALL WITH COMMON HEADER
655 002150* 042767 020000 175650 BIC #20000,STAT ;ABORTING SELECTION OF THIS MODULE
656 002156* 104400 000000* TEXT,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
657 ;
658 ;
659 ;
660 ;
661 ;
662 ;
663 ;
664 ;
665 ;
666 ;
667 ;
668 ;
669 ;
670 ;
671 ;
672 ;
673 ;
674 ;

```

```

675 002272* 005302          DEC      R2          ;UPDATE POINTER TO DATA BUFF
676 002274* 111267          MOVVB   (R2),AWAS   ;BAD DATA BYTE
677 002300* 005202          INC      R2          ;UPDATE POINTER TO DATA BUFF
678 002302* 112767          MOVVB   CHECKR,ASR ;GOOD DATA BYTE
679 002310* 005267          INC      CHECKR     ;UPDATE FOR NEXT TEST
680 002311* 005367          DEC      COUNT      ;ONE MORE BYTE HAS BEEN TESTED
681 002312* 005367          INC      COUNT      ;*****
682 002320* 104404 000000*  ;*****
683 002320* 104404 000000*  ;*****
684 002320* 104404 000000*  ;*****
685 002320* 104404 000000*  ;*****
686 002320* 104404 000000*  ;*****
687 002324* 005767 000572  RESTOR: TST      COUNT      ;ARE WE DONE CHECKING DATA ON
688 002330* 001253          BNE     CONTNU     ;ON THIS LINE
689 002332* 005267          INC      NODVTS     ;NOG GO DO THE REST OF THIS LINE
690 002332* 005267          INC      NODVTS     ;YES THIS LINE DONE ADD 1 TO
691 002336* 012711 100777          MOV     #100777,(R1) ;NODVTS>NO. OF LINES TESTED
692 002344* 022767          CMP     #10,NODVTS ;HAVE ALL LINES BEEN TESTED
693 002352* 001401          BEQ     PASS       ;GO TO END PASS CODE
694 002352* 000623          BR     CHECK1      ;RETURN TO MONITOR
695
696 002354*          PASS:
697 002354* 005067          CLR     TMOFLG     ;SHOW TIMER ROUTINE ITS TIME FOR ENDT
698 002360* 104400 000000*  EXITS,BEGIN        ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
002364* 000010          DULIN: .BLKW 8.          ;0
002404* 000010          DVAD1: .BLKW 8.          ;HIGH BYTE=SYNC COUNT NO.
002424* 000010          LNSYN1: .BLKW 8.         ;LOW BYTE=BINARY WORD PATTERN
002444* 000010          LNCNT1: .BLKW 8.        ;HIGH BYTE=NO. XMTD INTERRUPTS
002464* 000010          VRFLG1: .BLKW 8.        ;LOW BYTE = NO. RCV. INTERRUPTS
                                ;BYTE OFFSET VALUE FOR READ
;RECEIVE DATA 16 BYTES PER.BUFFER
002504* 000010          DULIN1: .BLKW 8.         ;DUI1 LINE #1 RECEIVE
                                ; DATA BUFFER
002524* 000010          DULIN2: .BLKW 8.         ;DUI1 LINE #2 RECEIVE
                                ; DATA BUFFER
002544* 000010          DULIN3: .BLKW 8.         ;DUI1 LINE #3 RECEIVE
                                ; TRANSMIT DATA BUFFER.
002564* 000010          DULIN4: .BLKW 8.         ;DUI1 LINE #4 RECEIVE
                                ; DATA BUFFER
002604* 000010          DULIN5: .BLKW 8.         ;DUI1 LINE #5 RECEIVE
                                ; DATA BUFFER
002624* 000010          DULIN6: .BLKW 8.         ;DUI1 LINE #6 RECEIVE
                                ; DATA BUFFER
002644* 000010          DULIN7: .BLKW 8.         ;DUI1 LINE #7 RECEIVE

```

```

731 002664* 000010          DULIN8: .BLKW 8.         ; DATA BUFFER
732 002664* 000010          ;DUI1 LINE #8 RECEIVE
733 002664* 000010          ; DATA BUFFER
734
735
736
737
738 002704* 000020          TQ:      .BLKW 16.        ;SOFTWARE SILO
739 002744* 000000          QAD1:   OPEN          ;Q POINTER
740 002746* 000000          QAD2:   OPEN          ;Q POINTER
741
742
743
744
745
746 002750* 004567 176212  LNKTAB: JSR     R5,DURCV   ;ANSWER LINE 1 RCV INTR
747 002752* 000000          0          ;OFFSET FOR LINE 1
748 002754* 004567 175776  JSR     R5,DUXMT   ;ANSWER LINE 1 XMT INTR
749 002756* 000000          0          ;OFFSET FOR LINE 1
750 002764* 004567 176176  JSR     R5,DURCV   ;ANSWER LINE 2 RCV INTR
751 002766* 000000          0          ;OFFSET FOR LINE 2
752 002770* 000000          2          ;ANSWER LINE 2 XMT INTR
753 002772* 004567 175762  JSR     R5,DUXMT   ;OFFSET FOR LINE 2
754 002774* 000000          2          ;ANSWER LINE 2 XMT INTR
755 002776* 004567 176162  JSR     R5,DURCV   ;ANSWER LINE 3 RCV INTR
756 002778* 000000          4          ;OFFSET FOR LINE 3
757 002780* 004567 175746  JSR     R5,DUXMT   ;ANSWER LINE 3 XMT INTR
758 002782* 000000          4          ;OFFSET FOR LINE 3
759 002784* 004567 176146  JSR     R5,DURCV   ;ANSWER LINE 4 RCV INTR
760 002786* 000000          6          ;OFFSET FOR LINE 4
761 002788* 004567 175732  JSR     R5,DUXMT   ;ANSWER LINE 4 XMT INTR
762 002790* 000000          6          ;OFFSET FOR LINE 4
763 002792* 004567 176132  JSR     R5,DURCV   ;ANSWER LINE 5 RCV INTR
764 002794* 000000          10         ;OFFSET FOR LINE 5
765 002796* 004567 175716  JSR     R5,DUXMT   ;ANSWER LINE 5 XMT INTR
766 002798* 000000          10         ;OFFSET FOR LINE 5
767 002800* 004567 176116  JSR     R5,DURCV   ;ANSWER LINE 6 RCV INTR
768 002802* 000000          12         ;OFFSET FOR LINE 6
769 002804* 004567 175702  JSR     R5,DUXMT   ;ANSWER LINE 6 XMT INTR
770 002806* 000000          12         ;OFFSET FOR LINE 6
771 002808* 004567 176102  JSR     R5,DURCV   ;ANSWER LINE 7 RCV INTR
772 002810* 000014          14         ;OFFSET FOR LINE 7
773 002812* 004567 175666  JSR     R5,DUXMT   ;ANSWER LINE 7 XMT INTR
774 002814* 000014          14         ;OFFSET FOR LINE 7
775 002816* 004567 176066  JSR     R5,DURCV   ;ANSWER LINE 8 RCV INTR
776 002818* 000016          16         ;OFFSET FOR LINE 8
777 002820* 000016          16         ;OFFSET FOR LINE 8
778
779 003110* 000000          PNTR:   OPEN          ;PNTR REG TO TEST DEVICE ON LINE
780 003112* 000026          TSYNC:  26          ;SYNC CHAR
781 003114* 000000          SELECT: OPEN        ;POINTER FLAG WHICH WILL BRANCH TO
782 003116* 000000          NODVTS: OPEN        ;TEST STATUS OF ALL LINES AFTER
783 003118* 000000          PASCT: OPEN        ;COMPLETING ONE LINE DATA TRANSFER
784 003120* 000000          ;WHEN =8 ALL LINES HAVE BEEN TESTED
785 003122* 000000          ;DATA BUFFER REG
786 003124* 000000          ;COUNTS LOOPS FOR WHEN TO ENDPASS

```





RCV DAT	003124R	489*	490	495	501	506	511	788#													
RCV RTN	001722R	569	574	580	594#																
REACT	= 004000	179#	443#	562																	
RESTOR	002324R	686#																			
RESTR	000270R	299#	530	541	552																
RESTR	000162R	79#	295	298#	415																
RES1	= 000056R	262#																			
RES2	= 000060R	263#																			
RING	= 040000	176#																			
RINTE	= 001104R	184#	385																		
RSTR	000112R	279#																			
RTS	= 000004	188#																			
RXC SR	= 000000	169#	385*	443	474*	496*	562	577*	601	603*											
RXDBUF	= 000002	170#	489																		
RXDDNE	= 000200	183#																			
RXERR	= 100000	192#																			
SADDR	000102R	294#																			
SELECT	003114R	294#	305	358	371	781#															
SEND	= 000020	219#	377																		
SETUP1	= 000416R	310	328#																		
SEVEN	= 004000	205#																			
SIX	= 002000	204#																			
SOF CNT	= 000042R	255#																			
SOPERS	= 104406	291#	536	547																	
SOPPAS	000046R	223#																			
SPOINT	= 000000	224#																			
SPSTZ	= 000040	11#	284																		
SRD	= 002000	180#																			
SR1	= 000016R	245#	378																		
SR2	= 000016R	245#																			
SR3	= 000022R	246#																			
SR4	= 000024R	247#																			
START	000248R	325#	291#																		
START2	= 000500R	324#																			
STAT	000026R	249#	655*																		
STD	= 000010	187#	385																		
STPSVN	= 000400	182#																			
SVR0	= 000066R	264#																			
SVR1	= 000064R	265#																			
SVR2	= 000066R	266#																			
SVR3	= 000070R	267#																			
SVR4	= 000072R	268#																			
SVR5	= 000074R	269#																			
SVR6	= 000076R	270#																			
SYNFX	= 020000	201#																			
SYNIN	= 030000	201#	380																		
SYNSCH	= 000020	186#	385																		
SYSCNT	000052R	259#																			
SVST	= 014000	377																			
TEXT	= 002000	65#	659#																		
TFLAG	003154R	298#	399*	800#																	
TICKER	003152R	402#	396*																		
TMOFI	003146R	406#	799#																		
TMCNT	003150R	401*	698*	797#																	
TMPSET	000654R	374	396#																		

TQ	002704R	303	304	423	425	439	441	738#														
TRCNT	003134R	293*	447*	792#																		
TRPDFD	= 000022	291#																				
TSVNC	= 003112R	457	780#																			
TXCSR	= 000004	173#	387*	454*	576*	602*																
TXDBUF	= 000006	116#	448*																			
TXDDNE	= 000200	116#																				
TXINTE	= 000100	217#	387																			
USER	= 000000	223#																				
VECTOR	000016R	240#	306																			
VOID	= 000001	190#																				
VRPLG1	002464R	565	571*	713#																		
WASADR	000104R	272#																				
WDR	000142R	281#	292*																			
WDT0	000114R	280#	291*																			
XFLAG	000005R	238#																				
XMRTH	= 001162R	469#	461#																			
.	003156R	738#	709#	711#	713#	718#	720#	722#	724#	726#	728#	730#	732#									

. ABS. 000000 000  
 003156 001

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0  
 XDUAIO, XDUAIO/SOL/CRF:SYM=DDXCOM, XDUAIO  
 RUN-TIME: 5 SECONDS  
 RUN-TIME RATIO: 14/4=2.9  
 CORE USED: 7K (13 PAGES)