

DUAI DEC/X11 SYSTEM EXERCISER MODULE
XDUAI0.P11 12-OCT-78 11:57

MACY11 30A(1052) 12-OCT-78 16:33 PAGE 2

SEQ 0001

.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 PATTERN. THE BINARY COUNT PATTERN USING THE MAINTENANCE MODE FEATURE. THE ERRORS 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: 923
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: SRI = 1 = ASYNCHRONOUS (ISOCRONOUS)
SRI = 0 = SYNCHRONOUS

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

12-OCT-78 11:57

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=DEVIBIT
;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=2000  
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=040  
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  
RINTE=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
```

```
OVERRUN=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
```

```

227 ;LIST SEQ BIN
228 ; SET UP VECTOR (RETURN ADDRESS(PC)) PC = INTR SERV. AREA.
229
230 IQMOD <DUAI> 6,1,5,5,2000,32
231 MODULE 140000 DUAI DEC/X11 SYSTEM EXERCISER MODULE
232 TITLE DUAI DEC/X11 SYSTEM EXERCISER MODULE
233 DDXCOM VERSION 6 23-MAY-78
234
235 *****LIST BIN*****
236 BEGIN:
237 MODNAM: .ASCII /DUAI / ;MODULE NAME.
238 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WRUFF USAGE
239 ADDR: .WORD ;1ST DEVICE ADDR.
240 VECOR: .WORD ;1ST DEVICE VECTOR.
241 BR1: .BYTE PRTV5+0 ;1ST BR LEVEL.
242 BR2: .BYTE PRTV5+0 ;2ND BR LEVEL.
243 DIVD1: .WORD ;DEVICE INDICATOR 1.
244 SR1: .OPEN ;SWITCH REGISTER 1.
245 SR2: .OPEN ;SWITCH REGISTER 2.
246 SR3: .OPEN ;SWITCH REGISTER 3.
247 SR4: .OPEN ;SWITCH REGISTER 4.
248 *****LIST BIN*****
249 STAT: 140000 ;STATUS WORD.
250 INIT: START ;MODULE START ADDR.
251 SPOINT: MODSP ;MODULE STACK POINTER.
252 PASCNT: MODSP ;PASS COUNTER.
253 ICOUNT: 2000 ;# OF ITERATIONS PER PASS=2000
254 SFCNT: 0 ;LOC TO COUNT ITERATIONS
255 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
256 SRFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
257 HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
258 SFCRCS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
259 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
260 CONFIC: 0 ;RESERVED FOR MONITOR USE
261 RES1: 0 ;RESERVED FOR MONITOR USE
262 SVRO: .OPEN ;LOC TO SAVE R0.
263 SVR1: .OPEN ;LOC TO SAVE R1.
264 SVR2: .OPEN ;LOC TO SAVE R2.
265 SVR3: .OPEN ;LOC TO SAVE R3.
266 SVR4: .OPEN ;LOC TO SAVE R4.
267 SVR5: .OPEN ;LOC TO SAVE R5.
268 SVR6: .OPEN ;LOC TO SAVE R6.
269 CSRA: .OPEN ;ADDR OF CURRENT CSR.
270 SBADR: .OPEN ;ADDR OF GOOD DATA, OR
271 ACSR: .OPEN ;CONTENTS OF CSR.
272 WASADR: .OPEN ;ADDR OF BAD DATA, OR
273 ASAT: .OPEN ;STATUS OF CONTENTS.
274 ERRTP: .OPEN ;TYPE OF ERROR.
275 ASB: .OPEN ;EXPECTED DATA.
276 ANA: .OPEN ;ACTUAL DATA.
277 RSTR: .RESTR ;RESTART ADDRESS AFTER END OF PASS
278 WDT0: .OPEN ;WORDS TO MEMORY PER ITERATION
279 WDFR: .OPEN ;WCORS FROM MEMORY PER ITERATION
280 INTR: .OPEN ;# OF INTERRUPTS PER ITERATION

```

```

283 IDNUM: 32 ;MODULE IDENTIFICATION NUMBER=32
284 .REPT SPSIZ ;MODULE STACK STARTS HERE.
285 .LIST 0
286 .ENDR
287
288 MODSP: *****LIST BIN*****
289
290 START: MOV #8,WDT0 ;8 WORDS TO MEM/ITERATION
291 MOV #3,WDFR ;8 WORDS FROM MEM/ITERATION
292 MOV #32,INTR ;32 INTERRUPTS/ITERATION
293 MOV #0,VIDI SELECT ;COPY THE DEVICE SELECTION PARAMETER
294 BNE RSTR ;ARE ANY SELECTED? IF YES, START PROCESSING
295 ;IF NO,DROP THE MODULE
296
297 RESTR: ENDS,BEGIN ;ONLY START W.D. TIMER ONCE
298 CLR #1,TFLAG ;RESET THE COUNT OF TRANSMITTER INTERRUPTS
299 CLR #1,RCCNT ;RESET THE RECEIVER COUNT
300 CLR DUDH ;CLR FLAG INDICATING NO. OF DU'S DONE
301 CLR NO,DU ;NO DU'S SELECTED
302 MOV #0,QAD1 ;SET UP Q POINTERS
303 MOV #0,QAD2 ;" " "
304 MOV #0,SELECT,R0 ;COPY ACTIVE SELECTION PARAMETER
305 MOV #LNKTA6,R2 ;R1 = VECTOR ADDRESS
306 ASR #1 ;R2 = LINK: USB TABLE WITH OFFSET
307 RCS #0 ;ISOLATE A SELECTION FLAG IN THE CARRY BIT
308 BEQ SETUP1 ;IF SELECTED GO SET UP VECTORS
309 ADD #10,R1 ;IF NO MORE,CONTINUE PROCESSING
310 ADD #14,R2 ;IF MORE,UPDATE POINTERS FOR NEXT DEVICE
311 BR #15 ;GO PROCESS NEXT DU11
312 INC R2,(R1)+ ;SET UP VECTOR RETURN ADDRESS(RCV)
313 MOV #0,R1 ;SET UP VECTOR PRIORITY (RCV)
314 INC R1 ;INCR. POINTER
315 MOV #R2,(R1)+ ;UPDATE NEW LINK ADDRESS
316 MOV #0,R1 ;SET UP VECTOR RETURN ADDRESS (XMT)
317 MOV #0,R1 ;SET UP VECTOR PRIORITY (XMT)
318 INC R1 ;INCR. POINTER
319 ADD #R2,R1 ;UPDATE NEW LINK ADDRESS FOR DUXMT
320 BR #15 ;GO CHECK FOR MORE DEVICES
321
322 ; THIS CODE WILL CLEAR ALL OF THE WRITE BUFFER AREA
323
324 SETUP1: MOV #103,R0 ;COUNT REQUIRED TO GO THRU
325 MOV #DULIN,R3 ;ALL DATA STORAGE BUFFERS
326 CLR (R3)+ ;STARTING ADDRESS OF
327 DEC R0 ;DATA BUFFER LOCATIONS.
328 BNE IS ;CLEAR DATA BUFF REG
329 ;ARE THERE MORE TO CLEAR?
330 ;NO GO BACK & DO THE REST
331
332 ; THIS CODE WILL SELECT WHICH LINES (<1:8>) HAVE
333 ; BEEN SELECTED FOR TEST & TRANSMIT SYNC TO START
334 ; 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 DULL  
342 000444* 012702 002404* MOV #DVAL1,R2 ;R2=ADDRESS OF TABLE OF ADDRESSES  
343 000450* 012703 002364* MOV #DULIN,R3 ;R3=ADDRESS OF TABLE OF BUFFER ADDRESSES  
344 000460* 012704 002304* MOV #DULIN,R4 ;POINT TO THE FIRST BUFFER  
345 000460* 010122 000000 1$:MOV R1,(R2)+ ;DVALX=DEVICE ADDR. CODE  
346 000462* 010423 000000 R4,(R3)+ ;BUFF POINTER HAS START OF LINE  
347 ;BUFF STORAGE  
348 000464* 062701 000010 ADD #10,R1 ;UPDATE  
349 000470* 062704 000020 ADD #20,R4 ;CNT DOWN  
350 000474* 005300 000000 DEC R0 ;NOT DONE GO BACK FOR MORE  
351 000476* 001370 000000 BNE 1$  
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 ASR R2 ;TEST IS THIS DEVICE ON LINE  
360 000530* 103406 BCS 4$ ;TEST GO SETUP MORE  
361 000532* 001414 BEQ ACTIVATE ;GO START ACTIVE DEVICES  
362 000534* 005113 000002 1$:COB (R3) ;SET LINCNT FLAG IF DEVICE NOT SELECTED  
363 000536* 005113 000002 ADD #2,R0 ;UPDATE THE TABLE POINTERS  
364 000542* 005723 TST (R3)+  
365 000544* 000770 BR DS ;TRY THE NEXT DEVICE  
366 000546* 112710 000004 4$:MOV #4,(R0) ;LOAD SYNC COUNT IN LISVN X(HIGH BYTE)  
367 000552* 012713 010420 MOV #10420,(R3) ;COUNT-1, HIGH FOR TRANSMIT (#OF CHAR. PLUS 1)  
368 ;LOW FOR RECEIVE BIT 3=TRANSMIT SYNC ON INTERRUPT  
369 000556* 005367 002334 DEC NODVTS ;KEEP TRACK OF NUMBER OF UNTESTED DEVICES  
370 000562* 000765 002324 BR 1$ ;TRY NEXT DEVICE  
371 000564* 000765 002324 ACTIVATE:MOV SELECT,R2 ;GET ACTIVE DEVICE FLAGS  
372 000570* 006202 000000 3$:MSR R2 ;ISOLATE A SELECTION FLAG  
373 000572* 103404 BCS 5$ ;IF SET, GO START DEVICE  
374 000574* 001421 BEQ TMRSET ;IF ALL SET, GO START TIMER  
375 000576* 001094 ADD #10,R1 ;TRY NEXT CSR  
376 000602* 000772 BR 6$ ;GO START NEXT DEVICE  
377 000604* 012761 014020 5$:MOV #SYSTST1SEND,TXCSR(R1) ;SET MAINT MODE & SEND  
378 000612* 005767 177200 TST SR1 ;ARE YOU RUNNING SYNCHRONOUSLY?  
379 000616* 001094 BNE 1$ ;NO, IT'S ISCHROMOUS  
380 000620* 012761 036026 000002 MOV #SYNNTIEIGHTINOPAR126,PARCSR(R1) ;SET SYNC INTERNAL,  
381 ;EIGHT BITS PER CHAR,NO PARITY ,26=SYNC  
382 000626* 000403 BR 2$ ;BRANCH AROUND  
383 000630* 012761 006026 000002 1$:MOV #SYNMODIEIGHTINOPAR126,PARCSR(R1) ;SET ISCHROMOUS MODE,  
384 ;EIGHT BITS PER CHAR,NO PARITY,26= SYNC  
385 000636* 012761 000520 000000 2$:MOV #RINTEN1STPSYN15VNSCH,RXCSR(R1) ;SET REC INTERRUPT ENABLE,  
386 ;STRIP SYNC SEARCH SYNC  
387 000644* 052761 000100 000004 BIS #TXINTE,TXCSR(R1) ;ENABLE TRANSMITTER INTERRUPT  
388 000652* 000751 BR 6$  
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 PRESCRIBED 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$:MOV TFLAG ;DON'T COME THRU HERE AGAIN  
400 000672* 012767 177777 002246 MOV #1,TMOTFLG ;SET THE TIMEOUT FLAG  
401 000700* 012767 000005 002242 MOV #2,TRCNT ;SET ENOUGH TIME FOR INTERRUPTS  
402 000706* 005067 002240 CLR TICKER ;START TIMEOUT  
403  
404 000712* 104407 000000* 1$: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 BEQ FINISH ;IF YES, GO FINISH UP  
408 000730* 005367 002216 DEC TICKER ;IF NO, TICK SOME TIME  
409 000734* 001366 BNE 1$ ;BREAK AGAIN,IF NO TIMEOUT  
410 000736* 104403 MSGNS,BEGIN,HUNG ;ASCII MESSAGE CALL WITH COMMON HEADER  
411 000744* 104410 ENDS,BEGIN ;  
412 000750* 000000* FINISH: ENDDTS,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 ADD #2,QAD1 ;UPDATE THE QUEUE POINTER  
423 000772* 022767 002744* 001752 CMP #16+40,QAD1 ;POINTER AT END OF QUEUE?  
424 001000* 001003 BNE 1$ ;BR IF NOT  
425 001002* 012767 002704* 001734 1$:MOV #1,QAD1 ;RESET THE POINTER  
426 001010* 012605 MOV (R6),R5 ;RESTORE THE OTHER GUY'S R5  
427  
428 001012* 000004 000000* 001020* ;-----  
429 ;IRQS,BEGIN,QOUT ; QUEUE UP TO CONTINUE AT QOUT AND RTI  
430 ;-----  
431  
432 ;DEFERRED XMTR SERVICE - THIS ROUTINE RETRIEVES POINTER TO CSR ADDRESS  
433 ;FROM THE FIFO QUEUE AND SERVICES THE LINE AT LEVEL 0  
434  
435  
436  
437 001020* 017700 001722 QOUT: MOV @QAD2,R0 ;GET POINTER FROM THE QUEUE  
438 001024* 062767 000002 ADD #2,QAD2 ;UPDATE THE QUEUE POINTER  
439 001032* 022767 002744* 001706 CMP #16+40,QAD2 ;POINTER AT HIGH LIMIT  
440 001040* 001003 BNE 1$ ;BR IF NOT  
441 001042* 012767 002704* 001676 MOV #1,QAD2 ;RESET THE POINTER  
442 001050* 016001 002494* 000000 MOV #DVAL1,R1 ;R1 = R0 WITH OFFSET VALUE  
443 001054* 032761 004000 1$:BIT #REACT,RXCSR(R1) ;TEST TO GIVE MORE SYNC CHARS  
444 001062* 001426 BEQ DUXMT2 ;BRANCH IF IN SYNC  
445 001064* 105360 002445* DECB LNCNT1+1(R0) ;SENT 16 CHARACTERS?  
446 001070* 001416 BEQ DUXMT3 ;ES-BRANCH  
447 001072* 005267 002036 INC TRCNT ;INCR TRANSMIT COUNT  
448 001076* 116061 002424* 000006 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 000000 DUXMT4: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
456 001140 104400 000000 DUXMT2: MOVB TSYNC,TXDBUF(R1) ;XMT SYNC CHAR (TSYNC)
457 001146 116761 001746 000006 DECB LNSYN1+1(R0) ;DEC SYNC COUNTER
458 001154 105360 002425 BNE XMTRTN ;EXIT IF SYNC COUNT NOT ZERO
459 001158 001000 000004 002425 XMTRTN: MOVB #4,LNSYN1+1(R0) ;RE-INITIALIZE SYNC COUNTER
460 001162 000000 000000 XMTRTN: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
461 001166 104400 000000 ; REQUEST FOR SERVICE
462 001170 010246 DURCV: MOV R2,-(SP) ;SAVE REG 2 ON STACK
463 001174 010346 MOV R3,-(SP) ;SAVE REG. 3 ON STACK
464 001178 010446 MOV R4,-(SP) ;SAVE REG. 4 ON STACK
465 001182 011503 MOV (R5),R3 ;GET OFFSET
466 001186 016304 002404 MOV DVAD1(R3),R4 ;R3 = R4 DEVICE CODE OFFSET VALUE
467 001190 105714 TSTB (R4) ;IS DURE SET
468 001194 100425 BNE DURCV1 ;DURE SET -SERV DONE REQUEST
469 001198 010467 176666 MOV R4,CSRA ;SHOW CSR ADDR.
470 001202 105714 BNE DURCV1 ;CONTENTS OF CSR
471 001206 010467 176666 MOV (R4),ACSR ;CONTENTS OF CSR
472 001210 011467 176664 CLR RXCSR(R4) ;RESTORE STACK
473 001214 005064 000000 MOV (SP)+,R4 ;
474 001218 005064 000000 MOV (SP)+,R3 ;
475 001222 012603 MOV (SP)+,R2 ;
476 001226 012602 MOV (SP)+,R5 ;
477 001230 012605 MOV (SP)+,R5 ;
478 001234 000000 000000 001240 PIRQS,BEGIN,FIRER ; QUEUE UP TO CONTINUE AT FIRER AND RTI
479 001238 000000 000000 001240 ;
480 001242 012767 000011 176640 FIRER: MOV #11,ERRTYP
481 001246 104405 000000 000000 HDRER$,BEGIN,NULL ;FALSE INTERRUPT
482 001250 000167 000662 JMP DROP ;TURN OFF THIS DEVICE
483 001254 016467 000002 001636 DURCV1: MOV RXDBUF(R4),RCVDAT ;SAVE RXDRUF
484 001258 005767 001632 TST RCVDAT ;ARE THERE ANY ERRORS ?
485 001262 100145 BRL #NO ;
486 001266 105363 002424 INCB LNSYN1(R3) ;UPDATE XMT DATA
487 001270 105263 002445 INCB LNCNT1+1(R3) ; " " ACTIVE COUNT
488 001274 010467 176570 MOV R4,CSRA ;CSR ADDR.
489 001278 010467 001610 MOV (R4),ACSR ;CONTENTS CSR
490 001282 005064 000000 CLR RXCSR(R4) ;RESTORE STACK
491 001286 012604 MOV (SP)+,R4 ;
492 001290 012603 MOV (SP)+,R3 ;
493 001294 012602 MOV (SP)+,R2 ;
494 001298 012605 MOV (SP)+,R5 ;
495 001302 032767 040000 001564 BIT #OVRERR,RCVDAT ;TEST FOR OVERRUN
496 001306 001403 BEQ IS ;NO
497 001310 000004 000000 001412 PIRQS,BEGIN,OVERR ; QUEUE UP TO CONTINUE AT OVERR AND RTI
498 001314 000000 000000 001412 ;
499 001318 032767 020000 001546 1$: BIT #FRMERR,RCVDAT ;TEST FOR FRAME ERROR
500 001322 000000 000000 001546 ;
501 001326 000000 000000 001546 ;
502 001330 000000 000000 001546 ;
503 001334 000000 000000 001546 ;
504 001338 000000 000000 001546 ;
505 001342 000000 000000 001546 ;
506 001346 000000 000000 001546 ;
```

```
507 001356 001403 BEQ 2$ ;NO
508 001360 000004 000000 001456 PIRQS,BEGIN,FRERR ; QUEUE UP TO CONTINUE AT FRERR AND RTI
509 001364 000000 000000 001456 ;
510 001368 032767 010000 001530 2$: BIT #PARER,RCVDAT ;TEST FOR PARITY ERROR
511 001372 001403 BEQ 3$ ;NO
512 001376 000004 000000 001522 PIRQS,BEGIN,PAERR ; QUEUE UP TO CONTINUE AT PAERR AND RTI
513 001380 000000 000000 001522 ;
514 001384 000004 000000 001566 3$: PIRQS,BEGIN,EBIT15 ; QUEUE UP TO CONTINUE AT EBIT15 AND RTI
515 001388 000000 000000 001566 ;
516 001392 000000 000000 001566 ;
517 001396 000000 000000 001566 ;
518 001400 000000 000000 001566 ;
519 001404 000000 000000 001566 ;
520 001408 000000 000000 001566 ;
521 001412 012767 000021 176466 OVERR: MOV #21,ERRTYP
522 001416 104405 000000 000000 HDRER$,BEGIN,NULL ;OVERRUN ERROR
523 001420 005267 001506 INCB OVERRU ;COUNT ERRORS
524 001424 022767 000003 001500 CMP #3,OVERRU ;3 ERRORS ?
525 001428 001403 BEQ IS ;YES
526 001432 000167 JMP RESTR ;NO RETRY MODULE
527 001436 005067 001466 1$: CLR OVERRU ;CLEAN UP
528 001440 000167 000464 JMP DROP ;GO Deselect THIS MODULE
529 001444 000167 000464 ;
530 001448 000167 000464 ;
531 001452 000167 000464 ;
532 001456 012767 000022 176422 FRERR: MOV #22,ERRTYP
533 001460 104406 000000 000000 HDRER$,BEGIN,NULL ;FRAME ERROR
534 001464 005267 001444 INCB FRAMER ;COUNT ERRORS
535 001468 022767 000003 001436 CMP #3,FRAMER ;3 ERRORS ?
536 001472 001403 BEQ IS ;YES
537 001476 000167 JMP RESTR ;NO RETRY MODULE
538 001480 005067 001424 1$: CLR FRAMER ;CLEAN UP
539 001484 000167 000420 JMP DROP ;GO Deselect THIS MODULE
540 001488 000167 000420 ;
541 001492 000167 000420 ;
542 001496 000167 000420 ;
543 001500 000167 000420 ;
544 001504 000167 000420 ;
545 001508 000167 000420 ;
546 001512 000167 000420 ;
547 001516 000167 000420 ;
548 001520 012767 000023 176356 PAERR: MOV #23,ERRTYP
549 001524 104406 000000 000000 HDRER$,BEGIN,NULL ;PARITY ERROR
550 001528 005267 001402 INCB PARTER ;COUNT ERRORS
551 001532 022767 000003 001374 CMP #3,PARTER ;3 ERRORS ?
552 001536 001403 BEQ IS ;YES
553 001540 000167 JMP RESTR ;NO RETRY MODULE
554 001544 005067 001362 1$: CLR PARTER ;CLEAN UP
555 001548 000167 000354 JMP DROP ;GO Deselect THIS MODULE
556 001552 000167 000354 ;
557 001556 000167 000354 ;
558 001560 000167 000354 ;
559 001564 000167 000354 ;
560 001568 012767 000000 176312 EBIT15: MOV #0,ERRTYP
561 001572 104405 000000 000000 HDRER$,BEGIN,NULL ;RECEIVE FLAG ERROR
562 001576 000167 000334 JMP DROP ;DESELECT THIS MODULE
563 001580 000167 000334 ;
564 001584 000167 000334 ;
565 001588 000167 000334 ;
566 001592 000167 000334 ;
567 001596 000167 000334 ;
568 001600 032764 004000 000000 READ: BIT #REACT,RXCSR(R4) ;IS DEVICE ACTIVE
569 001604 000000 000000 000000 ;
570 001608 000000 000000 000000 ;
571 001612 000000 000000 000000 ;
572 001616 000000 000000 000000 ;
```

```

563 001614* 001447 BEQ BADDONE ;GET OUT DEVICE NOT READY
564 001616* 005002 CLR R2 ;CLEAR BYTE PNTER
565 001620* 066302 002464* ADD VRFLG1(R3),R2 ;GET BYTE OFFSET
566 001624* 066302 002364* ADD DULIN(R3),R2 ;ADDR=DATA BUFF ADDR
567 001630* 066302 000002 MOVB 2(R4),R2 ;DATA => DATA BUFF
568 001634* 122712 000026 CMPB #2,(R2) ;SKP IF SYNC BIT
569 001640* 001430 BEQ RCVRTN ;
570 001646* 052267 001270 INC RCVRTN ;
571 001646* 052267 002464* INC VRFLG1(R3) ;
572 001652* 105363 002444* DECB LNCNT1(R3) ;
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 001670* 005267 001236 DUDN ;MASTER RESET (DEVICE)
580 001702* 001007 CMP NO,DU,DUDN ;COUNT THIS DU
581 001704* 012604 BNE RCVRTN ;ARE WE DONE CHCK'ING
582 001704* 012604 MOV (SP)+,R4 ;RESTORE STACK
583 001710* 012602 MOV (SP)+,R3 ;
584 001712* 012605 MOV (SP)+,R2 ;
585 ; ;
586 ; ;
587 001714* 000004 000000* 002014* ;IRQS,BEGIN,CHCK ; QUEUE UP TO CONTINUE AT CHCK AND RTI
588 ;-----;
589 ; ;
590 ; ;
591 ; ;
592 ; ;
593 ; ;
594 ; ;
595 001722* 012604 RCVRTN: MOV (SP)+,R4 ;RESTORE STACK POINTER
596 001724* 012603 MOV (SP)+,R3 ;
597 001726* 012602 MOV (SP)+,R2 ;
598 001732* 000002 RTI (SP)+,R5 ;
599 ; ; ;RETURN TO MAINLINE
600 ; ;
601 001734* 010467 176140 BADDONE: MOV MOV R4,CSRA ;SAVE CSR ADDRESS
602 001740* 005064 000004 176134 MOV RXCSR(R4),ACSR ;CONFERS OF CSR
603 001752* 005064 000000 CLR TXCSR(R4) ;COLLAPSE THIS DEVICE IT A'INT WORKIN'
604 001756* 012604 CLR RXCSR(R4) ;
605 001760* 012603 MOV (SP)+,R4 ;RESTORE STACK POINTER
606 001764* 012605 MOV (SP)+,R3 ;
607 001764* 012605 MOV (SP)+,R2 ;
608 ; ;
609 001766* 000004 000000* 001774* ;IRQS,BEGIN,BDONE ; QUEUE UP TO CONTINUE AT BDONE AND RTI
610 ;-----;
611 ; ;
612 001774* 012767 000011 176104 BDONE: MOV #11,ERRRTP
613 ;*****;
614 002002* 104405 000000* 000000 ;RDERS,BEGIN,NULL ;YOU SHOULD NOT HAVE INTERRUPTED WITHOUT RECACT ASSEFTED
615 ;*****;
616 002010* 000167 000126 JMP DRUP ;DESELECT THIS MODULE
617 ;
618 ;

```

```

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 002030* 005002 CLR R2 ;CLR BUFF POINTER
626 002032* 012701 002444* 1S: MOV #LNCNT1,R1
627 002036* 105711 TSTB (R1)
628 002040* 001402 BEQ CHCK2
629 002042* 001072 CMP (R1),R2
630 002044* 000774 BR 1S
631 002046* 010200 CHCK2: MOV R2,R0 ;R0 WILL HOLD LINE NO./2
632 002050* 012767 007364* MOV DULIN(R2),R2 ;R2=START ADDR. THIS LINE BUFF
633 002054* 012767 001046 MOV R0,R1 ;GET FIRST CHAR.
634 002060* 126722 001042 CONTNU: CMPB CHECKR,(R2)+ ;CHECK DATA & INCR. POINTER
635 002064* 001410 BEQ 1S ;THIS WORD GOOD GO CHECK MORE
636 002066* 122767 000026 001032 CMPB #26,CHECKR ;WAS IT SYNC CHAR.
637 002070* 001073 BNE ERRRT ;NO GO REPORT ERROR
638 002076* 005267 001024 INC CHECKR ;YES UPDATE CHECKR
639 002102* 005302 DEC R2 ;UPDATE DULIN BUFFER POINTER
640 002104* 000767 BR CONTNU ;GO BACK & CHECK REAL DATA
641 002106* 000767 1S: INC CHECKR ;SET UP FOR NEXT BYTE TEST
642 002112* 005367 001004 DEC COUNT ;ONE MORE BYTE HAS BEEN TESTED
643 002116* 001360 BNE CONTNU ;NOT DONE YET GO CHECK MORE
644 002120* 005267 000772 INC NODVTS ;THIS LINE DONE ADD 1 TO
645 ; ; ;NG. OF DEVICES TESTED
646 002124* 012711 100777 MOV #100777,(R1)
647 002130* 022767 000010 000760 CMP #10,NODVTS ;HAVE ALL LINES BEEN TESTFD
648 002136* 001506 BEQ PASS ;GO TO END PASS CODING
649 002140* 000730 BR CHCK1
650 ;
651 ;
652 ;
653 002142* DROP:
654 002142* 104403 000000* 002162* MSGNS,BEGIN,1S ;ASCII MESSAGE CALL WITH COMMON HEADER
655 002150* 042767 020000 175650 BIC #20000,STAT ;ABORTING SELECTION OF THIS MODULE
656 002156* 044400 000000* TEXT,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
657 002162* 002166* 1S: TEXT ;MESSAGE POINTER
658 002164* 177777 ;" TERMINATOR
659 002166* 020040 042504 044526 TEXT: .ASCIZ " DEVICE FAILED TO INTERRUPT DROPPING MODULE "
660 002174* 042503 043040 044501
661 002207* 042514 020104 047524
662 002210* 044440 052116 051105
663 002216* 052222 052120 042040
664 002220* 050120 047110 047111
665 002232* 020107 047515 052504
666 002240* 042514 000040
667 ;
668 002244* 042504 044526 042503 MESSAG2: .ASCIZ "DEVICE HUNG"
669 002250* 044040 047125 000107 HUNG: MESSAG2
670 002260* 002244* 177777
671 002262* 177777
672 ;
673 ;
674 002264* 016067 002404* 175606 ERRRT: MOV DVAD1(R0),CSRA ;CSRA=LINE ADDR.

```

```
675 002272* 005302          DEC R2          ;UPDATE POINTER TO DATA BUFF
676 002274* 111267          MOVB (R2),AWAS ;BAD DATA BYTE
677 002300* 005202          INC R2          ;UPDATE POINTER TO DATA BUFF
678 002302* 112767          MOVB 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          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          NODVTS       ;YES THIS LINE DONE ADD 1 TO
691 002336* 012711 100777          MOV #100777,(R1) ;NODVTS>NO. OF LINES TESTED
692 002342* 022767          CMP #10,NODVTS ;HAVE ALL LINES BEEN TESTED
693 002352* 001401          BEQ PASS     ;GO TO END PASS CODE
694 002352* 000623          BR CHCK1    ;RETURN TO MONITOR
695
696 002354*          PASS:          CLR TMOFLG    ;SHOW TIMER ROUTINE ITS TIME FOR ENDT
697 002360* 104400 000000*  EXITS,BEGIN  ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
698
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. ;DU11 LINE #1 RECEIVE
                          ; DATA BUFFER
002524* 000010          DULIN2: .BLKW 8. ;DU11 LINE #2 RECEIVE
                          ; DATA BUFFER
002544* 000010          DULIN3: .BLKW 8. ;DU11 LINE #3 RECEIVE
                          ; TRANSMIT DATA BUFFER.
002564* 000010          DULIN4: .BLKW 8. ;DU11 LINE #4 RECEIVE
                          ; DATA BUFFER
002604* 000010          DULIN5: .BLKW 8. ;DU11 LINE #5 RECEIVE
                          ; DATA BUFFER
002624* 000010          DULIN6: .BLKW 8. ;DU11 LINE #6 RECEIVE
                          ; DATA BUFFER
002644* 000010          DULIN7: .BLKW 8. ;DU11 LINE #7 RECEIVE
```

```
731 002664* 000010          DULIN8: .BLKW 8. ; DATA BUFFER
732 002664* 000010          ;DU11 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 002762* 004567 176176  JSR R5,DURCV ;ANSWER LINE 2 RCV INTR
751 002764* 000000          2 ;OFFSET FOR LINE 2
752 002766* 004567 175762  JSR R5,DUXMT ;ANSWER LINE 2 XMT INTR
753 002768* 000000          2 ;OFFSET FOR LINE 2
754 002774* 004567 176162  JSR R5,DURCV ;ANSWER LINE 3 RCV INTR
755 002776* 000000          4 ;OFFSET FOR LINE 3
756 002778* 004567 175746  JSR R5,DUXMT ;ANSWER LINE 3 XMT INTR
757 002780* 000000          4 ;OFFSET FOR LINE 3
758 002786* 004567 176146  JSR R5,DURCV ;ANSWER LINE 4 RCV INTR
759 002788* 000000          6 ;OFFSET FOR LINE 4
760 002790* 004567 175732  JSR R5,DUXMT ;ANSWER LINE 4 XMT INTR
761 002792* 000000          6 ;OFFSET FOR LINE 4
762 002798* 004567 176132  JSR R5,DURCV ;ANSWER LINE 5 RCV INTR
763 002800* 000010          10 ;OFFSET FOR LINE 5
764 002802* 004567 175716  JSR R5,DUXMT ;ANSWER LINE 5 XMT INTR
765 002804* 000010          10 ;OFFSET FOR LINE 5
766 002810* 004567 176116  JSR R5,DURCV ;ANSWER LINE 6 RCV INTR
767 002812* 000012          12 ;OFFSET FOR LINE 6
768 002814* 004567 175702  JSR R5,DUXMT ;ANSWER LINE 6 XMT INTR
769 002816* 000012          12 ;OFFSET FOR LINE 6
770 002822* 004567 176102  JSR R5,DURCV ;ANSWER LINE 7 RCV INTR
771 002824* 000014          14 ;OFFSET FOR LINE 7
772 002826* 004567 175666  JSR R5,DUXMT ;ANSWER LINE 7 XMT INTR
773 002828* 000014          14 ;OFFSET FOR LINE 7
774 002834* 004567 176066  JSR R5,DURCV ;ANSWER LINE 8 RCV INTR
775 002836* 000016          16 ;OFFSET FOR LINE 8
776 002838* 004567 175652  JSR R5,DUXMT ;ANSWER LINE 8 XMT INTR
777 002840* 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          ;COMPLETING ONE LINE DATA TRANSFER
784 003120* 000000          ;WHEN=8 ALL LINES HAVE BEEN TESTED
785 003122* 000000          ;DATA BUFFER REG
786 003124* 000000          PASCT: OPEN ;COUNTS LOOPS FOR WHEN TO ENDPASS
```


XDUAI0.P11 12-OCT-78 11:57 CROSS REFERENCE TABLE -- USER SYMBOLS

RCV DAT	003124R	489*	490	495	501	506	511	788#													
RCV RTN	001722R	569	574	580	594#																
REACT=	004000	489#	443#	562																	
RESTOR	002324R	686#																			
RESTR	000270R	299#	530	541	552																
RESTRT	000162R	799#	295	298#	415																
RES1	000056R	262#																			
RES2	000060R	263#																			
RING	= 040000	176#																			
RINTEH	= 001104	184#	385																		
RSTRT	= 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#																			
SOPCMT	= 000042R	255#																			
SOPERS	= 104406	291#	536	547																	
SOPPAS	= 000046R	223#																			
SPOINT	= 000022R	223#																			
SPSTZ	= 000040	1	284																		
SRD	= 002000	180#																			
SR1	= 000016R	244#	378																		
SR2	= 000016R	245#																			
SR3	= 000022R	246#																			
SR4	= 000024R	247#																			
START	= 000248R	250#	291#																		
START2	= 000500R	325#																			
STAT	= 000026R	249#	655*																		
STD	= 000010	187#	385																		
STPSVN	= 000400	182#																			
SVR0	= 000062R	265#																			
SVR1	= 000064R	265#																			
SVR2	= 000066R	266#																			
SVR3	= 000070R	267#																			
SVR4	= 000072R	268#																			
SVR5	= 000074R	269#																			
SVR6	= 000076R	270#																			
SYNFXI	= 020000	204#																			
SYNINT	= 030000	204#	380																		
SYNSCH	= 000020	186#	385																		
SYSCNT	= 000052R	259#																			
SYSTST	= 014000	377																			
TEXT	= 000066R	65#	659#																		
TFLAG	= 003154R	298#	399*	800#																	
TICKER	= 003152R	402#	396*																		
TMOFF	= 003146R	406#	408*																		
TMCNT	= 003150R	401*	799#																		
TMPSET	= 000654R	374	396#	696*	797#																

XDUAI0.P11 12-OCT-78 11:57 CROSS REFERENCE TABLE -- USER SYMBOLS

TQ	002704R	303	304	423	425	439	441	738#															
TRCNT	003134R	293*	447*	792#																			
TRPDFD	= 000022	291#																					
TSVNC	= 003112R	457	780#																				
TXCSR	= 000002	171#	387*	457*	454*	576*	602*																
TXDBUF	= 000006	116#	448*																				
TXDDNE	= 000200	217#	387																				
TXINTE	= 000100	217#																					
USER	= 000000	220#	306																				
VECTOR	= 00016R	240#																					
VOID	= 000001	190#																					
VRPLG1	= 002464R	565	571*	713#																			
WASADR	= 000104R	272#																					
WDR	= 00014R	281#	292*																				
WDTO	= 000114R	280#	291*																				
XFLAG	= 000005R	238#																					
XMRTH	= 001182R	469	461#	709#	711#	713#	718#	720#	722#	724#	726#	728#	730#	732#									
.	= 003156R	738#	708#																				

. ABS. 000000 000
003156 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

XDUAI0,XDUAI0/SOL/CRF:SYM=DDXCOM,XDUAI0
RUN-TIME: 5 SECONDS
RUN-TIME RATIO: 14/4=2.9
CORE USED: 7K (13 PAGES)