```
        .REM
```

IDENTIFICATION

PRODUCT CODE: AC-E905B-MC

PRODUCT NAME: CXVTAB0 VT20 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

VTA IS AN IOMOD THAT EXERCISES UP TO FOUR VT20'S (8 DL11 LINES). IT IS INTENDED TO BE A DATA HANDLING ROUTINE USED IN CONJUNCTION WITH TEST 21 OF MAINDEC-11-DBVTA (PREVIOUSLY LOADED AND RUNNING IN THE VT20'S PDP11-05). DATA IS ENTERED AT EACH SELECTED TUBE AND SET INTO THE CONTINUOUS TRANSMIT MODE. THIS DATA IS THEN RECEIVED AND RETRANSMITTED BY THE VT20 HOST COMPUTER (THIS DEC/X11 MODULE). ALL LINES SELECTED FOR TEST (UP TO 8 DL11'S WITH CONTIGUOUS ADDRESSES AND VECTORS) CAN BE ACTIVATED AND RUN CONCURRENTLY. ALL TRANSMIT AND RECEIVE ERRORS ARE REPORTED ON THE CONSOLE TTY. NO DATA ERRORS ARE REPORTED BY THIS MODULE.

2. REQUIREMENTS

HARDWARE: AT LEAST ONE VT20

STORAGE:: VTA REQUIRES:
1. DECIMAL WORDS: 2296
2. OCTAL WORDS: 04370
3. OCTAL BYTES: 10760

3. PASS DEFINITION

ONE PASS OF THE VTA MODULE CONSISTS OF CONTINUOUSLY RECEIVING AND TRANSMITTING THE DATA ENTERED ON ALL SELECTED LINES FOR THE PERIOD DEFINED BELOW.

4. EXECUTION TIME

EXECUTION TIME VARIES WITH THE NUMBER OF JOBS (MODULES) ACTIVE, THE BAUD RATE AND THE NUMBER OF TUBES BEING EXERCISED. HOWEVER, THIS MODULE RUNNING ALONE WILL TAKE ABOUT 20 SECONDS FOR ONE TUBE AND UP TO 60 SECONDS FOR 8 TUBES. BAUD RATES LESS THAN 2400 REQUIRE PROGRAM MODIFICATION WHICH INCREASES EXECUTION TIME. SEE SECTION 6.B.

5. CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA:175610, VCT:340, BR1:5, BR2:0, DVC:1

REQUIRED PARAMETERS:

VCT:VECTOR ADDRESS OF FIRST DL11 IF NOT 340
DVC:NO OF DL11'S (TUBES) IF GREATER THAN 1

6. DEVICE SETUP

A. THE USER MUST LOAD AND START TEST 21 OF MAINDEC-11-DBVTA IN THE VT20 PDP11/05 IN ORDER FOR THIS MODULE TO EXERCISE. CONSULT THE ABOVE DOCUMENT AND COMPLY WITH THE OPERATING INSTRUCTIONS FOR TEST 21 (SECTION 26). THIS DEC/X11 MODULE EXPECTS THE USER TO ENTER DATA ON EACH SELECTED TUBE AND SET EACH TUBE IN THE CONTINUOUS TRANSMIT MODE. THIS STEP IS TAKEN AFTER THE DEC/X11 EXERCISER HAS BEEN STARTED BY THE 'RUN' COMMAND. TYPICAL USER ACTION ON EACH SELECTED TUBE WILL BE AS FOLLOWS:

KEY	FUNCTION
CTRL E	CLEAR SCREEN
CTRL W	GENERATE WORST CASE CHARACTER PATTERN ON TOP OF SCREEN
CTRL T	CONTINUOUS TRANSMIT TO DEC/X11 MODULE (DEC/X11 MODULE WILL RECEIVE DATA AND TRANSMIT IT BACK TO BOTTOM OF SCREEN)

NOTE: IF THE CHARACTER PATTERN FAILS TO RETURN ON THE BOTTOM OF THE SCREEN AFTER ONE 'CTRL T', THEN RETRY AFTER 'END PASS' IS REPORTED FOR THIS DEC/X11 MODULE (DL11 RECIEVERS ARE TURNED OFF SECONDS BEFORE 'END PASS' MSG). IF DATA IS STILL NOT RETURNED FROM HOST COMPUTER (DEC/X11 SYSTEM) THEN VERIFY THE VT20 HOST COMPUTER BY RUNNING MAINDEC-11-DZVTE.

B. IF BAUDS RATES LOWER THAN 2400 ARE USED THEN ONE OF THE VALUES BELOW MUST BE PLUGGED IN LOCATION 'COUNT1' (VTA 1660) TO AVOID POSSIBLE 'DL11 HUNG' ERRORS. NOTE THAT EXECUTION TIME INCREASES BY ABOUT 30 SECONDS FOR EVERY 1 COUNT.

BAUD RATE	VALUE
1200	2
600	2
300	2
150	3
110	4

USE THE "MOD" COMMAND TO ALTER LOCATION 'COUNT1' (VTA 1660)

7. MODULE OPERATION

7.1 TEST SEQUENCE

A. START: USING THE DEVICE SELECTION PARAMETER "DVID1" THIS SECTION OF CODE SETS UP THE VECTORS OF ALL SELECTED LINES TO POINT TO THE APPROPRIATE JSR IN THE JSR LINKING TABLE.

B. SETCSR: THIS PIECE OF CODE INSERTS THE PROPER CSR ADDRESS OF EACH ACTIVE LINE INTO THE THIRD

WORD OF EACH JSR TABLE ENTRY.

C. SETUP: THIS CODE INITIALIZES ALL TABLES,BUFFERS,FLAGS, AND COUNTERS.

D. STRTUP: THIS CODE TURNS ON THE INTERRUPT ENABLES FOR EACH SELECTED RECEIVER.

E. TIMR: THIS CODE IS AN 'END PASS' TIMER LOOP VIA 'BREAKS' TO THE MONITOR. THE PROGRAM LEAVES THIS LOOP TO RESTART LINES ON TRANSMIT AFTER COMPLETION OF RECEIVING A BLOCK OF DATA, AND TO PREPARE FOR THE 'END PASS' MESSAGE AND TO REPORT RECEIVER ERRORS IF ANY.

F. XSTRT: RESTARTS EACH LINE TRANSMITTING THAT HAS RECEIVED A BLOCK OF DATA (CHARACTER '014' TERMINATES A BLOCK OF DATA).

G. TMOUT: THIS CODE IS ENTERED WHEN THE 'END PASS' MESSAGE IF CALLED FOR. IT PROVIDES TIME VIA 'BREAKS' FOR ALL LINES TO BECOME IDLE AT COMPLETION OF RECEIVE. REPORTS RECEIVER ERRORS IF ANY AND CHECKS THAT AT LEAST ONE GOOD LINE IS LEFT TO BE RESTARTED.

H. HNGTST: AFTER WAITING ENOUGH TIME FOR ALL LINES TO BECOME IDLE, ALL LINES ARE EXPECTED TO HAVE COMPLETED RECEIVING. IF NOT, THE LINE IS REPORTED HUNG AND DROPPED (SEE SECTION 6 FOR BAUD RATES LESS THAN 2400). 'END PASS' MESSAGE IS NOW TYPED.

I. RESTRT: THIS CODE INITIALIZES QUEUES AND TIMERS AND STARTS UP ANY LINE TO RECEIVE PREVIOUSLY REPORTED AS HUNG, THEN GOES AND STARTS UP ALL OTHER LINES TRANSMITTING.

J. RINT: THE RECEIVER SERVICE ROUTINE STORES DATA, AND CHECKS FOR RECEIVER ERRORS WHICH ARE STORED FOR BACKGROUND REPORTING. IT ALSO LOOKS FOR THE TERMINATING CHARACTER OF '014'. WHEN RECEIVED, IT SETS UP FOR TRANSMIT.

K. TINT: THE TRANSMITTER SERVICE ROUTINE SIMPLY QUEUES UP THE REQUEST FOR SERVICE IN A FIFO QUEUE, UPDATES THE QUEUE POINTER, AND RETURNS CONTROL BACK TO THE MONITOR WITH A 'PIRQ'. THE ELEMENT THAT GETS STORED IN THE QUEUE IS A POINTER TO THE INTERRUPTING CSR ADDRESS. THE ACTUAL SERVICING IS DONE LATER WHERE THE SERVICE CODE IS EXECUTED AT LEVEL 0.

L. TSERV: THIS CODE RETRIEVES A POINTER FROM THE FIFO QUEUE AND BUILDS THE CSR ADDRESS. STATUS IS CHECKED AND ERRORS REPORTED. IF THE NEXT CHARACTER IS THE TERMINATING CODE (014) THE LINE WILL THEN BE INITIALIZED

TO RECEIVE. IN ANY EVENT, THE NEXT CHARACTER WILL BE OUTPUTTED AND AN EXIT BACK TO THE MONITOR PREFORMED.

M. RERCK: THIS CODE REPORTS ANY RECEIVER ERRORS WHICH HAD BEEN DUMPED INTO THE ERROR QUEUE 'EQ'.

7.2 DESCRIPTION OF TABLES, QUEUES, AND BUFFERS

A. DLSTUS: 8 WORD TABLE WHICH SPECIFIES WHAT EACH LINE IS DOING: NOT SELECTED (0), TRANSMITTING (1), RECEIVING (2), SELECTED AND WAITING FOR USER ACTION ON VT20 KEYBOARD (3), WAITING FOR TRANSMIT AFTER RECEIVE (-1). ALL LINES SHOULD REACH THIS STATE BEFORE 'END PASS'.

B. RCVSW: 8 WORD TABLE INDICATING THAT A LINE HAS FOUND THE SYNC OR START CODE OF '377' WHICH BEGINS A BLOCK OF DATA: 0 = NO SYNC, 1 = SYNC HAS BEEN MADE.

C. ERTAB: 8 WORD TABLE WHICH INDICATES THE NUMBER OF ERRORS THAT HAVE OCCURRED DURING EACH TRANSFER.

D. TQ: 8 WORD FIFO QUEUE FOR TRANSMITTER SERVICE. LOADED WITH A POINTER TO THE CSR ADDRESS AND UNLOADED DURING DEFERRED XMTR SERVICE.

E. EQ: 48 WORD FIFO QUEUE FOR RECEIVER ERROR REPORTING. LOADED WITH THE BAD LINE'S CSR ADDRESS AND STATUS AND DATA - 2 WORDS PER ERROR.

F. XRB0-7: 8-390 BYTE RECEIVE/TRANSMIT DATA BUFFERS

G. JSRTAB: 64 WORD TABLE THAT CONTAINS 16 JSR INSTRUCTINS WITH TWO TRAILING ARGUMENTS. EACH RECEIVER AND EACH XMITR HAS AN ASSIGNED JSR IN THE TABLE OF THE FOLLOWING FORMAT:

```
        JSR     R5,RINT(TINT)
        0
        N
```

WHERE THE 0 GETS OVERLAYED WITH THE ADDRESS OF THE CSR FOR LINE N AND N IS THE LINE NO. IN OCTAL TIMES TWO (00-16)

8. OPERATOR OPTIONS

A. THE USER CAN MODIFY (VTA 14) "DVID1" TC SELECT OR DESELECT INDIVIDUAL DL11'S.

B. THE USER CAN USE THE "MOD" COMMAND TO DUMP THE TABLES OR BUFFERS DESCRIBED IN 7.2 TO OBTAIN MORE DETAILED ERROR INFORMATION.

9. ERROR PRINTOUTS

9.1 ERROR FORMAT - RECEIVE

CSRA = CSR ADDRESS
CSRC = DBR WORD AS FOLLOWS:

BIT 15 = DL11 ERROR
**BIT 14 = OVERRUN
BIT 13 = FRAMING
BIT 12 = PARITY
BIT 7-0= DATA RECEIVED

OCTAL WORD FOLLOWING ERROR DEFINED AS FOLLOWS:

BIT 3 = ILLEGAL INTERRUPT - INT EN OR DONE NOT SET
BIT 2 = ILLEGAL SYNC CHARACTER - COULD NOT SYNC ON RECEIVE (1ST NON-ZERO CHAR WAS NOT THE #377 CODE)
BIT 1 = DL11 DROPPED FROM MODULE - 3 ERRORS OCCURRED ON ANY DATA BLOCK
BIT 0 = DL11 IS HUNG - DL FAILED TO RECEIVE A BLOCK OF DATA IN WORST CASE TIME.

**NOTE: OVERRUN ERRORS WILL START OCCURING ON LARGE SYSTEMS WHERE BUS ACTIVITY IS HIGH AS THE NUMBER OF VT20 TUBES ACTIVATED IS INCREASED. I.E. IF 8 TUBES AT 9600 BAUD ARE SELECTED, THE WORSTCASE RECEIVER INTERRUPT SERVICE TIME IS APPROXIMATELY 140 MICRO-SECONDS. THIS MEANS THAT IF ALL OTHER I/O BUS ACTIVITY PLUS SOFTWARE SLOP INHIBITS A RECEIVER INTERRUPT FROM BEING SERVICED IN 140 MICRO-SECONDS THEN RECEIVER OVERRUN ERRORS START OCURRING AT THE DL11 LINE ELECTRICALLY MOST DISTANT FROM THE PROCESSOR.

9.2 ERROR FORMAT - TRANSMIT

CSRA = CSR ADDRESS
CSRC = CSR CONTENTS AS FOLLOWS:

BIT 7 = XMITR READY
BIT 6 = XMITR INTERRUPT ENABLED

OCTAL WORD FOLLOWING ERROR DEFINED AS FOLLOWS:

BIT 1 = XMITR IS HUNG - DL FAILED TO INTERRUPT
BIT 0 = DL11 DROPPED FROM MODULE - OCCURES ON ABOVE ERROR

-

```
                                        IOMOD   <VTAB >,175610,340,5,,,2,64
                                        MODULE  140000,VTAB ,175610,340,5,,,2,64
                                        .TITLE  VTAB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION   6       23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************
                                BEGIN:
000000  052126  041101     040 MODNAM: .ASCII  /VTAB / ;MODULE NAME.
000005      000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006  175610                  ADDR:   175610+0                ;1ST DEVICE ADDR.
000010  000340                  VECTOR: 340+0                   ;1ST DEVICE VECTOR.
000012      240                 BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013      000                 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: MODSP                   ;MODULE STACK POINTER.
000034  000000                  PASCNT: 0                       ;PASS COUNTER.
000036  000002                  ICONT:  2                       ;# OF ITERATIONS PER PASS=2
000040  000000                  ICOUNT: 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                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050  000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052  000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054  000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056                          CONFIG:                         ;RESERVED FOR MONITOR USE
000056  000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060  000000                  RES2:   0                       ;RESERVED FOR MONITOR USE
000062  000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064  000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066  000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
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                          SBADR:                          ;ADDR OF GOOD DATA, OR
000102  000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104                          WASADR:                         ;ADDR OF BAD DATA, OR
000104  000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106                          ERRTYP:                         ;TYPE OF ERROR
000106  000000                  ASB:    OPEN                    ;EXPECTED DATA.
000110  000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112  001204'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114  000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116  000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120  000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122  000064                  IDNUM:  64                      ;MODULE IDENTIFICATION NUMBER=64
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                        .NLIST
```



```
                                        .WORD   0
                                        .LIST
                                        .ENDR
000224'                         MODSP:
                                ;*****************************************************************
                                ;THIS ROUTINE SETS UP THE VECTORS FOR ALL SELECTED LINES TO POINT
                                ;TO THE APPROPRIATE JSR IN THE JSR LINK TABLE.

000224' 012767  000606  177666  START:  MOV     #390.,INTR              ;390 INTERRUPTS/ITERATION
000232' 012767  000606  177654          MOV     #390.,WDTO              ;390 WORDS TO MEM/ITERATION
000240' 016700  177544                  MOV     VECTOR,R0       ;SET R0 TO POINT TO THE 1ST VECTOR
000244' 016701  177544                  MOV     DVID1,R1        ;LOAD R1 WITH DEVICE SELECTION PARMETER
000250' 012702  010400'                 MOV     #JSRTAB,R2      ;SETUP R2 TO POINT TO JSR TABLE
000254' 006201                  1$:     ASR     R1              ;SHIFT SELECT BIT INTO "C"
000256' 103026                          BCC     3$              ;BR IF NOT SELECTED
000260' 062767  000606  177626          ADD     #390.,WDTO              ;390 MORE
000266' 062767  000606  177624          ADD     #390.,INTR              ;390 MORE INTS.
000274' 010220                          MOV     R2,(R0)+        ;SET UP RCVR INTR POINTER
000276' 116720  177510                  MOVB    BR1,(R0)+       ;SET UP RCVR PRIORITY LEVEL
000302' 005200                          INC     R0              ;MOVE POINTER
000304' 062702  000010                  ADD     #10,R2          ;POINT R2 TO XMTR ENTRY IN JSRTAB
000310' 010220                          MOV     R2,(R0)+        ;SET UP XMTR INTR POINTER
000312' 116720  177474                  MOVB    BR1,(R0)+       ;SET UP XMTR PRIORITY LEVEL
000316' 005200                          INC     R0              ;MOVE POINTER
000320' 062702  000010                  ADD     #10,R2          ;POINT R2 TO RCVR ENTRY FOR NEXT LINE
000324' 022702  010600'         2$:     CMP     #SETQTS,R2      ;IS THE POINTER AT THE END OF THE TABLE?
000330' 001351                          BNE     1$              ;BR IF NOT
000332' 000405                          BR      SETCSR          ;GO SET UP CSR ADDRESSES
000334' 062700  000010          3$:     ADD     #10,R0          ;ADVANCE VECTOR POINTER
000340' 062702  000020                  ADD     #20,R2          ;ADVANCE JSR TABLE POINTER
000344' 000767                          BR      2$              ;GO CHECK FOR END OF TABLE

                                ;THIS ROUTINE SETS UP THE JSR TABLE SUCH THAT THE APPROPRIATE
                                ;CSR ADDRESS IS INCLUDED AS THE 3RD WORD OF EACH ENTRY

000346' 016700  177434  SETCSR: MOV     ADDR,R0         ;GET THE FIRST CSR ADDRESS INTO R0
000352' 016701  177436          MOV     DVID1,R1        ;LOAD R1 WITH THE DEVICE SELECTION PARAMETER
000356' 001002                  BNE     1$              ;BR IF SOMETHING SELECTED
000360' 104410  000000'         END$,BEGIN              ;DROP MODULE-NOTHING SELECTED
000364' 012702  010404'         1$:     MOV     #JSRTAB+4,R2    ;POINT R2 TO CSR ADDRESS ENTRY
000370' 006201                  2$:     ASR     R1              ;SHIFT SELECT BIT INTO "C"
000372' 103016                          BCC     4$              ;BR IF LINE NOT SELECTED
000374' 010012                          MOV     R0,(R2)         ;PUT RCVR CSR ADDRESS IN TABLE
000376' 062700  000004                  ADD     #4,R0           ;GENERATE XMITR CSR ADRS IN R0
000402' 062702  000010                  ADD     #10,R2          ;POINT TO XMTR SLOT IN JSR TABLE
000406' 010012                          MOV     R0,(R2)         ;PUT XMTR CSR ADDRESS IN THE TABLE
000410' 062700  000004                  ADD     #4,R0           ;GENERATE RCVR CSR ADRS IN R0
000414' 062702  000010                  ADD     #10,R2          ;POINT TO RCVR SLOT IN JSR TABLE
000420' 022702  010604'         3$:     CMP     #SETQTS+4,R2    ;IS POINTER BEYOND END OF TABLE?
000424' 001361                          BNE     2$              ;BR IF NOT
000426' 000405                          BR      SETUP           ;GO SETUP FOR RECEIVE
000430' 062700  000010          4$:     ADD     #10,R0          ;UPDATE CSR ADDRESS
000434' 062702  000020                  ADD     #20,R2          ;UPDATE JSR TABLE POINTER
000440' 000767                          BR      3$              ;GO TEST FOR END OF TABLE

                                ;THIS CODE CLEARS BUFFERS AND TABLES AND INITIALIZES FLAGS
```

```
                                ;
000442' 005000                  SETUP:  CLR     R0              ;
000444' 016060  002040' 002014' 1$:     MOV     XRADO(R0),XRBPO(R0) ;SET UP BUFFER POINTER-RCVR
000452' 062760  000003  002014'         ADD     #3,XRBPO(R0)    ;ADVANCE POINTER OVER NULL CHARS
000460' 062700  000002                  ADD     #2,R0           ;ADVANCE TO NEXT BUFFER
000464' 022700  000020                  CMP     #20,R0          ;HAVE WE SET UP 8 BUFFERS
000470' 001365                          BNE     1$              ;BR IF MORE
000472' 012700  002320'                 MOV     #XRB0,R0        ;SETUP 1ST ADRS OF 1ST BUFFER
000476' 005020                  2$:     CLR     (R0)+           ;CLEAR THIS LOCATION
000500' 022700  010400'                 CMP     #JSRTAB,R0      ;HAVE WE CLEARED ALL BUFFERS?
000504' 001374                          BNE     2$              ;BR IF NOT
000506' 012701  002100'                 MOV     #RCVSW,R1       ;SETUP 1ST RCVR SOFTWARE SWITCH ADRS
000512' 012700  002120'                 MOV     #ERTAB,R0       ;SETUP 1ST ERROR COUNTERS ADRS
000516' 005020                  3$:     CLR     (R0)+           ;CLR THIS COUNTER
000520' 005021                          CLR     (R1)+           ;CLR THIS RCVR SOFTWARE SWITCH ADRS
000522' 020027  002140'                 CMP     R0,#ERTAB+20    ;HAVE ALL ERROR COUNTERS BEEN CLRED
000526' 001373                          BNE     3$              ;BR IF NOT
000530' 004767  010044                  JSR     PC,SETQTS       ;GO SET UP Q'S & TIMERS
000534' 012700  177776                  MOV     #-2,R0          ;INITIALLY SET TO MINUS TWO
000540' 005001                          CLR     R1              ;CLR INITIAL DEVICE COUNT
000542' 016702  177246                  MOV     DVID1,R2        ;GET DEVICE SELECTION PARAMETER
000546' 062700  000002                  ADD     #2,R0           ;ADVANCE TABLE OFFSET
000552' 022700  000020                  CMP     #20,R0          ;HAVE LOOKED AT 8 DL'S?
000556' 001003                          BNE     5$              ;BR IF NOT
000560' 010167  001210                  MOV     R1,ACTDEV       ;SAVE THE COUNT OF ACTIVE LINES
000564' 000412                          BR      STRTUP          ;ACTIVE TABLE SET UP
000566' 006202                  5$:     ASR     R2              ;SHIFT SELECT BIT INTO "C"
000570' 103403                          BCS     6$              ;BR IF SELECTED
000572' 005060  002060'                 CLR     DLSTUS(R0)      ;THIS LINE IDLE
000576' 000763                          BR      4$              ;GO TEST NEXT LINE
000600' 005201                  6$:     INC     R1              ;COUNT THIS LINE
000602' 012760  000003  002060'         MOV     #3,DLSTUS(R0)   ;THIS LINE IS SELECTED & WAITING KEYBOARD ACTION
000610' 000756                          BR      4$              ;GO TEST NEXT LINE

                                ;THIS CODE STARTS UP ALL SELECTED LINES RECEIVING

000612' 016702  177170          STRTUP: MOV     ADDR,R2         ;GET STARTING CSR ADRS TO R2
000616' 010201                          MOV     R2,R1           ;GET STARTING CSR ADRS TO R1
000620' 062702  000100                  ADD     #100,R2         ;MAKE R2 LAST RCVR CSR ADRS + 10
000624' 012700  002060'                 MOV     #DLSTUS,R0      ;GET DL STATUS TABLE ADRS
000630' 005720                  1$:     TST     (R0)+           ;SEE IF THIS LINE SELECTED
000632' 001404                          BEQ     2$              ;BR IF NOT
000634' 005761  000002                  TST     2(R1)           ;FLUSH DONE BIT
000640' 052711  000100                  BIS     #100,(R1)       ;SET RCVR INT FOR THIS LINE
000644' 062701  000010          2$:     ADD     #10,R1          ;ADVANCE TO NEXT RCVR
000650' 020102                          CMP     R1,R2           ;HAVE WE DONE 8 LINES?
000652' 001366                          BNE     1$              ;BR IF NOT

                                ;THIS CODE DOES THE FOLLOWING:
                                ;RETURNS TO MONITOR VIA "BREAK" FOR "END PASS" TIMING
                                ;REPORTS ANY RECEIVER ERRORS IN THE ERROR QUEUE "EQ"
                                ;RESTARTS ALL DL LINES WAITING TO XMIT DATA

000654'                         TIMR:
000654' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000660' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```



```
000664' 005367  001106                  DEC     CNTR            ;COUNT EOP TIMER
000670' 001003                          BNE     1$              ;BR IF NOT DOE FOR "END PASS" MSG
000672' 005367  001102                  DEC     CNTR1           ;
000676' 001427                          BEQ     TMOUT           ;STAY IN LOOP TILL CNTR1 = 0
000700' 004767  007742          1$:     JSR     PC,RERCK        ;GO REPORT RECVR ERRORS IF ANY

000704' 016700  177076          XSTRT:  MOV     ADDR,R0         ;GET BASE DL ADRS
000710' 062700  000004                  ADD     #4,R0           ;POINT TO XMITR CSR
000714' 005001                          CLR     R1              ;ZERO OFFSET
000716' 005761  002060'         1$:     TST     DLSTUS(R1)      ;THIS LINE WAITING FOR XMIT?
000722' 100005                          BPL     2$              ;BR IF NOT
000724' 012761  000001  002060'         MOV     #1,DLSTUS(R1)   ;INDICATE NOW XMITING
000732' 012710  000100                  MOV     #100,(R0)       ;ENABLE THIS LINE'S XMIT INT ENABLE
000736' 062700  000010          2$:     ADD     #10,R0          ;ADVANCE TO NEXT XMITR'S CSR ADRS
000742' 062701  000002                  ADD     #2,R1           ;ADVANCE OFFSET FOR STATUS TABLE
000746' 022701  000020                  CMP     #20,R1          ;HAVE ALL LINES BEEN CHECKED
000752' 001361                          BNE     1$              ;BR IF NOT
000754' 000737                          BR      TIMR            ;RETURN TO EOP TIMING


                                ;THIS CODE DOES THE FOLLOWING:
                                ;ENTERED WHEN DUE FOR "END PASS" MSG AND ALLOWS TIME
                                ;(BREAKS) FOR ALL LINES TO COMPLETE RECEIVING
                                ;REPORTS ALL RECEIVER ERRORS IF ANY
                                ;DROPS MODULE IF ALL DL11 RCVR'S HAVE MADE 3 ERRORS

000756' 016767  001010  001014  TMOUT:  MOV     COUNT1,CNTR1    ;SET UP "TIMEOUT" TIMERS
000764' 012767  020000  001004          MOV     #20000,CNTR     ;
000772'                         1$:
000772' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000776' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001002' 005367  000770                  DEC     CNTR            ;COUNT FOR WORST CASE XMIT/RCV TIME
001006' 001371                          BNE     1$              ;BR IF NOT TIMED OUT
001010' 005367  000764                  DEC     CNTR1           ;
001014' 001366                          BNE     1$              ;STAY IN LOOP TILL CNTR1 = 0
001016' 004767  007624                  JSR     PC,RERCK        ;GO REPORT RCVR ERRORS IF ANY
001022' 005767  000746                  TST     ACTDEV          ;MAKE SURE SOMETHING IS ACTIVE
001026' 001002                          BNE     HNGTST          ;BR IF SO
001030' 104410  000000'                 END$,BEGIN              ;DROP MODULE - ALL SELECTED RCVERS MADE 3 ERS

                                ;THIS CODE REPORTS ANY DL LINE WHICH BECAME HUNG
                                ;DROPS MODULE IF LAST SELECTED LINE BECAME HUNG
                                ;TYPES THE "END PASS" MESSAGE

001034' 005000                  HNGTST: CLR     R0              ;ZERO OFFSET
001036' 016701  176744                  MOV     ADDR,R1         ;GET BASE DL ADRS
001042' 005760  002060'         1$:     TST     DLSTUS(R0)      ;SEE IF THIS LINE IS WAITING FOR XMIT
001046' 003445                          BLE     3$              ;BR IF IT IS OR IF NOT SELECTED
001050' 022760  000003  002060'         CMP     #3,DLSTUS(R0)   ;WAS IT SELECTED BUT NO KEYBOARD ACTION?
001056' 001441                          BEQ     3$              ;BR IF SO
001060' 012767  000003  000724          MOV     #3,STATUS       ;SET THE HUNG & DROPPED BITS IN THE STATUS WD
001066' 016167  000002  177006          MOV     2(R1),ACSR      ;GET RECEIVER CSR CONTENTS
001074' 010167  177000                  MOV     R1,CSRA         ;SET UP RCVR CSR ADRS
001100' 022760  000002  002060'         CMP     #2,DLSTUS(R0)   ;WAS IT RECEIVING?
001106' 001406                          BEQ     2$              ;BR IF SO
001110' 062767  000004  176762          ADD     #4,CSRA         ;SET UP XMIT CSR ADRS
```

```
001116' 017767  176756  176756          MOV     @CSRA,ACSR      ;GET XMITR CSR CONTENTS
001124' 005077  176750          2$:     CLR     @CSRA           ;DISABLE THIS LINE'S INTERRUPT ENABLE
001130' 012767  000023  176750          MOV     #23,ERRTYP      ;DEV DID NOT INTERRUPT
                                        ;*****************************************************
001136' 104405  000000' 002034'         ARDER$,BEGIN,STATBL     ;INDICATE THIS LINE IS HUNG & DROPPED
                                        ;*****************************************************
001144' 005060  002060'                 CLR     DLSTUS(R0)      ;DESELECT THIS LINE
001150' 005367  000620                  DEC     ACTDEV          ;SUB FROM TOTAL LINES ACTIVE
001154' 001002                          BNE     3$              ;BR IF MORE LINES ACTIVE
001156' 104410  000000'                 END$,BEGIN              ;DROP MODULE - LAST SELECTED LINE BECAME HUNG
001162' 062701  000010          3$:     ADD     #10,R1          ;ADVANCE TO NEXT DL BUS ADRS
001166' 062700  000002                  ADD     #2,R0           ;ADVANCE OFFSET
001172' 022700  000020                  CMP     #20,R0          ;HAVE WE TESTED ALL DL'S?
001176' 001321                          BNE     1$              ;BR IF NOT
001200' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS

                                ;THIS CODE WILL RESTART ANY DL LINE TO RECEIVE THAT WAS HUNG AT THE TIME OF
                                ;THE 'END PASS' - ALL OTHER DL'S WILL BE RESTARTED TRANSMITTING

001204' 004767  007370          RESTRT: JSR     PC,SETQTS       ;GO SET UP Q'S AND TIMERS
001210' 016700  176572                  MOV     ADDR,R0         ;GET BASE DL ADRS
001214' 005001                          CLR     R1              ;CLEAR OFFSET
001216' 005761  002060'         1$:     TST     DLSTUS(R1)      ;IS THIS LINE NOT SELECTED OR WAITING  XMIT?
001222' 003410                          BLE     2$              ;BR IF SO
001224' 016161  002040' 002014'         MOV     XRADO(R1),XRBPO(R1) ;SETUP RECEIVER POINTER
001232' 062761  000003  002014'         ADD     #3,XRBPO(R1)    ;SKIP POINTER OVER NULLS
001240' 012710  000100                  MOV     #100,(R0)       ;ENABLE THIS LINE'S RCVR INT ENABLE
001244' 062700  000010          2$:     ADD     #10,R0          ;ADVANCE TO NEXT DL BUS ADRS
001250' 062701  000002                  ADD     #2,R1           ;ADVANCE OFFSET
001254' 022701  000020                  CMP     #20,R1          ;HAVE ALL LINES BEEN CHECKED?
001260' 001356                          BNE     1$              ;BR IF NOT
001262' 000167  177416                  JMP     XSTRT           ;GO START ALL OTHER LINES TRANSMITTING

                                ;RECEIVER INTERRUPT SERVICE-ENTERED VIA APPROPRIATE JSR TABLE ENTRY
                                ;CHECKS STATUS AND STORES DATA INFORMATION
                                ;STORES RECVR ERRORS IN QUEUE 'EQ' FOR BACKGROUND REPORTING
                                ;SETS UP FOR XMIT ON END OF TRANSMISSION

001266' 010046                  RINT:   MOV     R0,-(SP)        ;SAVE R0,R1 & R2 ON THE STACK
001270' 010146                          MOV     R1,-(SP)
001272' 010246                          MOV     R2,-(SP)
001274' 012500                          MOV     (R5)+,R0        ;GET RCVR CSR ADRS TO R0
001276' 016001  000002                  MOV     2(R0),R1        ;GET RCVR DBF CONTENTS TO R1
001302' 011502                          MOV     (R5),R2         ;GET LINE # TIMES TWO TO R2
001304' 132710  000100                  BITB    #100,(R0)       ;ENABLE MUST BE UP
001310' 001003                          BNE     1$              ;BR IF OK
001312' 052701  004000                  BIS     #4000,R1        ;RECORD PHONEY INTERRUPT IN BIT 11
001316' 000427                          BR      5$              ;GO AND SAVE ERROR
001320' 005762  002100'         1$:     TST     RCVSW(R2)       ;HAVE WE BEEN RECEIVING ON THIS LINE?
001324' 001016                          BNE     4$              ;BR IF SO
001326' 122701  000377                  CMPB    #377,R1         ;IS THE CHAR EQUAL TO THE START CODE?
001332' 001411                          BEQ     3$              ;BR IF SO
001334' 105701                          TSTB    R1              ;IS THE CHAR A NULL?
001336' 001004                          BNE     2$              ;BR IF NOT
001340' 012762  000002  002060'         MOV     #2,DLSTUS(R2)   ;INDICATED THIS LINE NOW HAS KEYBOARD ACTION
```



```
001346' 000472                          BR      9$              ;GO RETURN FROM NULL CHAR INT
001350' 052701  002000          2$:     BIS     #2000,R1        ;RECORD BAD SYNC CODE
001354' 000410                          BR      5$              ;GO AND SAVE ERROR
001356' 005262  002100'         3$:     INC     RCVSW(R2)       ;RECORD THAT THIS LINE RECEIVED A START CODE
001362' 110172  002014'         4$:     MOVB    R1,@XRBPO(R2)   ;SAVE DATA BYTE
001366' 005262  002014'                 INC     XRBPO(R2)       ;ADVANCE BUFFER POINTER
001372' 005701                          TST     R1              ;LOOK FOR DL11 ERRORS
001374' 100041                          BPL     8$              ;BR IF NONE
001376' 026727  000404  002320' 5$:     CMP     ERPTR1,#XRB0    ;ERROR POINTER OVER END OF ERROR BUFFER?
001404' 001003                          BNE     6$              ;BR IF NOT
001406' 012767  002160' 000372          MOV     #EQ,ERPTR1      ;RESET ERROR QUEUE POINTER
001414' 010077  000366          6$:     MOV     R0,@ERPTR1      ;SAVE CSR ADRS OF BAD LINE
001420' 062767  000002  000360          ADD     #2,ERPTR1       ;ADVANCE ERROR BUFFER POINTER
001426' 005262  002120'                 INC     ER$AB(R2)       ;COUNT THIS ERROR
001432' 022762  000003  002120'         CMP     #3,ER$AB(R2)    ;IS THIS THE LAST ALLOWABLE ERROR?
001440' 001007                          BNE     7$              ;BR IF NOT
001442' 005010                          CLR     (R0)            ;DISABLE THIS LINE'S RCVR INT ENABLE
001444' 005062  002060'                 CLR     DLSTUS(R2)      ;DESELECT THIS LINE
001450' 052701  001000                  BIS     #1000,R1        ;SET UP INDICATION FOR MAX # OF ERRORS THIS LINE
001454' 005367  000314                  DEC     ACTDEV          ;SUBTRACT FROM TOTAL ACTIVE
001460' 010177  000322          7$:     MOV     R1,@ERPTR1      ;SAVE ERROR & DATA INFORMATION
001464' 062767  000002  000314          ADD     #2,ERPTR1       ;ADVANCE ERROR BUFFER POINTER
001472' 032701  006000                  BIT     #6000,R1        ;SEE HOW TO EXIT
001476' 001016                          BNE     9$              ;BR IF EITHER ILLEGAL INT OR SYNC ERROR
001500' 122701  000014          8$:     CMPB    #14,R1          ;SEE IF THE TERMINATING CHAR
001504' 001013                          BNE     9$              ;BR IF NOT
001506' 005062  002100'                 CLR     RCVSW(R2)       ;INDICATE RECVR IS IDLE
001512' 016262  002040' 002014'         MOV     XRADO(R2),XRBPO(R2) ;SET UP XMITR BUFFER POINTER
001520' 005062  002120'                 CLR     ER$AB(R2)       ;RESET THIS LINE ERROR COUNTER
001524' 005010                          CLR     (R0)            ;DISABLE THIS LINE'S RCVR INT ENABLE
001526' 012762  177777  002060'         MOV     #-1,DLSTUS(R2) ;REQUEST XMIT FOR THIS LINE
001534' 012602          9$:     MOV     (SP)+,R2        ;RESTORE THE OTHER GUY'S REGS
001536' 012601                  MOV     (SP)+,R1
001540' 012600                  MOV     (SP)+,R0
001542' 012605                  MOV     (SP)+,R5
001544' 000002                  RTI                     ;RETURN TO THE OTHER GUY

                                ;TRANSMITTER INTERRUPT SERVICE-ENTERED VIA APPROPRIATE JSR TABLE
                                ;ENTRY WITH R5 POINTING TO THE CSR ADDRESS-CONTENTS OF R5
                                ;GETS QUEUED UP IN FIFO QUEUE AND ROUTINE RETURNS CONTROL BACK TO
                                ;THE MONITOR VIA A 'PIRQ' TO DEFER SERVICING XMITTER AT LEVEL 0

001546' 010577  000230          TINT:   MOV     R5,@QPTR1       ;STORE CONTENTS OF R5 IN THE QUEUE
001552' 062767  000002  000222          ADD     #2,QPTR1        ;UPDATE THE QUEUE POINTER
001560' 022767  002160' 000214          CMP     #TQ+20,QPTR1    ;POINTER AT END OF QUEUE?
001566' 001003                          BNE     1$              ;BR IF NOT
001570' 012767  002140' 000204          MOV     #TQ,QPTR1       ;RESET THE POINTER
001576' 012605          1$:     MOV     (SP)+,R5        ;RESTORE THE OTHER GUY'S R5
                                        ;-----------------------------------------------------------
001600' 000004  000000' 001606'         PIRQ$,BEGIN,TSERV       ; QUEUE UP TO CONTINUE AT TSERV AND RTI
                                        ;-----------------------------------------------------------

                                ;DEFERRED XMTR SERVICE-THIS ROUTINE RETRIEVES POINTER TO CSR ADDRESS
                                ;FROM THE FIFO QUEUE AND SERVICES THE LINE AT LEVEL 0

001606' 017700  000172          TSERV:  MOV     @QPTR2,R0       ;GET POINTER FROM THE QUEUE
```

```
                                ;UPDATE THE QUEUE POINTER
                                ;POINTER AT HIGH LIMIT?
                                ;BR IF NOT
                                ;RESET THE POINTER
                                ;MOVE THE CSR ADRS TO R1
                                ;GET LINE # TO R0
                                ;GET CSR CONTENTS
                                ;XMITR DONE + ENABLE MUST BE UP
                                ;BR IF NOT
                                ;ADVANCE TO NEXT DATA
                                ;GET NEXT DATA BYTE
                                ;IS IT THE TERMINATING CHARACTER?
                                ;BR IF NOT
                                ;SET UP RCVR POINTER
                                ;SET RCVR POINTER TO START CODE
                                ;CLEAR XMITR INT ENABLE
                                ;ZERO ERROR COUNTER THIS LINE
                                ;INDICATE NOW RECEIVING
                                ;SET RCVR INT ENABLE
                                ;SEND NEXT DATA BYTE
                                ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                ;DISABLE XMITR INT ENABLE IF SET
                                ;SET UP CSR ADDRESS
                                ;SET UP CONTENTS OF CSR
                                ;ILLEGAL INTERRUPT
                                ;*****************************************************
                                ;XMTR ILLEGAL INT
                                ;*****************************************************
                                ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                                ;SOME POINTERS, VARIABLES AND CONSTANTS UNIQUE TO THIS MODULE

                                ;#OF 64K BREAKS TO MONITOR FOR 'END PASS'
                                ;#OF 64K BREAKS TO MONITOR FOR 'TIMOUT' TEST
                                ;#OF DL11S' ACTIVE
                                ;COUNTS BREAKS TO MONITOR
                                ;COUNTS GROUPS OF 64K BREAKS TO MONITOR
                                ;MULTI INT FIFO QUEUE POINTER-LOAD
                                ;MULTI INT FIFO QUEUE POINTER-UNLOAD
                                ;MULTI ERROR FIFO QUEUE POINTER-LOAD
                                ;MULTI ERROR FIFO QUEUE POINTER-UNLOAD
                                ;CONTAINS ADDITIONAL ERROR STATUS INFO

                                ;POINTER TO NEXT BYTE ADRS IN THE DATA
                                ;BUFFER ON BOTH RECEIVE & TRANSMIT

                                ;POINTER TO ER STATUS WD
                                ;PRINT OUT TERMINATOR

                                ;ADDRESSES OF THE 8 DL11
```



```
                                ;XMITR & RCVR DATA BUFFERS

                                ;TABLES, QUEUES & BUFFERS

                                ;LOCS SPECIFY STATUS OF EACH DL
                                ;LOCS SPECIFY (WHEN NON-ZERO) DL HAS RECD NULLS
                                ;RCVR ERROR COUNTERS - ONE PER DL
                                ;XMITR SERVICE FIFO QUEUE
                                ;48 WORD RCVR FIFO ERROR QUEUE

                                ;390 BYTE XMIT/RCV DATA BUFFERS
                                ;FOR 8 DL11 LINES

                                ;THE FIRST 3 BYTES ARE NULL CHARS
                                ;(0'S), THE REST OF EACH BUFFER WILL
                                ;RECEIVE A START CODE OF 377 AND UP TO
                                ;384 BYTES OF DATA, EACH DATA XFER IS
                                ;TERMINATED WITH THE CHAR '14'.

                                ;JSR LINK TABLE CONSISTING OF 16 JSR'S (8 RCVR + 8 XMTR)
                                ;THAT LINK THE INTERRUPTS TO COMMON SERVICE ROUTINES

                                ;RECEIVER LINE FOR LINE 0
                                ;SET UP WITH RCVR CSR ADRS
                                ;IDENTIFIES THIS LINE
                                ;TRANSMITTER LINE FOR LINE0
                                ;SET UP WITH XMTR CSR ADRS
                                ;IDENTIFIES THIS LINE
                                ;LINK FOR LINE 1
                                ;LINK FOR LINE 2
                                ;LINK FOR LINE 3
                                ;LINK FOR LINE 4
```

```
                                                        0
                                                        10
                                        JSR     R5,RINT         ;LINK FOR LINE 5
                                                        0
                                                        12
                                        JSR     R5,TINT
                                                        0
                                                        12
                                        JSR     R5,RINT         ;LINK FOR LINE 6
                                                        0
                                                        14
                                        JSR     R5,TINT
                                                        0
                                                        14
                                        JSR     R5,RINT         ;LINK FOR LINE 7
                                                        0
                                                        16
                                        JSR     R5,TINT
                                                        0
                                                        16

                                ;THIS ROUTINE SETS UP QUEUE POINTERS & TIMERS ON START & RESTART

                                SETQTS: MOV     #TQ,QPTR1       ;SET UP XMITR FIFO QUEUE POINTERS
                                        MOV     #TQ,QPTR2       )
                                        MOV     #EQ,ERPTR1      ;SET UP ERROR FIFO QUEUE POINTERS
                                        MOV     #EQ,ERPTR2      )
                                        MOV     COUNT,CNTR1     ;SET UP 'END PASS' COUNTERS
                                        MOV     #40000,CNTR     )
                                        RTS     PC              ;RETURN

                                ;THIS ROUTINE CHECKS FOR AND REPORTS ANY RECEIVE
                                ;ERRORS THAT HAVE BEEN DUMPED IN THE QUEUE (EQ)

                                RERCK:  CMP     ERPTR1,ERPTR2   ;ANY ERRORS PENDING?
                                        BEQ     2$              ;BR IF NOT
                                        MOV     ERPTR2,R0       ;GET ERROR POINTER
                                        MOV     (R0)+,CSRA      ;SET UP CSRA WITH DEVICE ADRS
                                        MOV     (R0)+,ACSR      ;SET UP ACSR WITH DBR STATUS INFO
                                        MOV     ACSR,STATUS     ;GET ERROR STATUS
                                        BIC     #7400,ACSR      ;SAVE ONLY DL11 DBR STATUS
                                        BIC     #170377,STATUS  ;SAVE ALL OTHER ERRORS IN 'STATUS'
                                        SWAB    STATUS          ;RIGHT JUSTIFY CONTENTS OF 'STATUS'
                                        MOV     R0,ERPTR2       ;SAVE UPDATED ERROR POINTER
                                        CMP     R0,#XRB0        ;IS ERROR POINTER AT END OF ERROR BUFFER?
                                        BNE     1$              ;BR IF NOT
                                        MOV     #EQ,ERPTR2      ;RESET ERROR POINTER
                                1$:     MOV     #17,ERRTYP      ;UNKNOWN RECEIVER ERROR
                                        ;*************************************************************
                                        HRDER$,BEGIN,STATBL     ;RECEIVER ERROR
                                        ;*************************************************************
                                        BR      RERCK           ;CHECK FOR MORE ERRORS IN QUEUE
                                2$:     RTS     PC              ;RETURN
                                        .END
```



```
ACSR    000102R             350#    526*    531*    666*    793*    794     795*
ACTDEV  001774R             443#    511     538*    606*    677#
ADDR    000006R             316#    398     455     481     520     552
ADDR22= 001000              314#
ASB     000106R             354#
ASTAT   000104R             352#
AWAS    000110R             355#
BEGIN   000000R             313#    401     473     474     504     505     513     535     540     545     636     663     669
                            671     804
BIT0  = 000001              368#
BIT1  = 000002              368#
BIT10 = 002000              368#
BIT11 = 004000              368#
BIT12 = 010000              368#
BIT13 = 020000              368#
BIT14 = 040000              368#
BIT15 = 100000              368#
BIT2  = 000004              368#
BIT3  = 000010              368#
BIT4  = 000020              368#
BIT5  = 000040              368#
BIT6  = 000100              368#
BIT7  = 000200              368#
BIT8  = 000400              368#
BIT9  = 001000              368#
BREAK$= 104407              368#    473     474     504     505
BR1     000012R             318#    381     385
BR2     000013R             319#
BTOD$ = 104421              368#
CDATA$= 104412              368#
CNTR    001776R             475*    502*    506*    678#    783*
CNTR1   002000R             477*    501*    508*    679#    782*
CONFIG  000056R             338#
COUNT   001770R             675#    782
COUNT1  001772R             501     676#
CSRA    000100R             348#    527*    530*    531     532*    665*    792*
DATCK$= 104411              368#
DATER$= 104404              368#
DLSTUS  002060R             447*    450*    458     484     486*    521     523     528     537*    554     586*    604*    617*
                            660*    709#
DVID1   000014R             320#    374     399     439
ENDIT$= 104413              368#    545
END$  = 104410              368#    401     513     540
EQ      002160R             597     713#    780     781     801
ERPTR1  002006R             595     597#    598*    599*    607*    608*    682#    780*    789
ERPTR2  002010R             683#    781*    789     791     798*    801*
ERRTYP  000106R             353#    533*    667*    802*
ERTAB   002120R             431     434     600*    601     615*    659*    711#
EXIT$ = 104400              368#    663     671
GETPA$= 104415              368#
GWBUF$= 104414              368#
HNGTST  001034R             512*    519#
HRDCNT  000044R             333#
HRDER$= 104405              368#    535     669     804
HRDPAS  000050R             335#
ICONT   000036R             330#
```

```
ICOUNT  000040R          331#
IDNUM   000122R          360#
INIT    000030R          327#
INTR    000120R          359#  371*  379*
JSRTAB  010400R          375   402   428   727#
MAP22$= 104416           368#
MODNAM  000000R          314#
MODSP   000224R          328   366#
MSGN$  = 104403          368#
MSGS$  = 104402          368#
MSG$   = 104401          368#
NULL   = 000000          368#  669
OPEN   = 000000          315   321   322   323   324   341   342   343   344   345   346   347   348
                         350   352   354   355   357   358   359   368#  680   681   682   683   686
                         687   688   689   690   691   692   693
OTOA$  = 104420          368#
PASCNT  000034R          329#
PIRQ$  = 000004          368#  636
POPSP  = 005726          368#
POPSP2= 022626           368#
PRTY   = 000000          319   368#
PRTY0  = 000000          368#
PRTY1  = 000040          368#
PRTY2  = 000100          368#
PRTY3  = 000140          368#
PRTY4  = 000200          368#
PRTY5  = 000240          318   368#
PRTY6  = 000300          368#
PRTY7  = 000340          368#
PS     = 177776          368#
PSW    = 177776          368#
PUSH   = 005746          368#
PUSH2  = 024646          368#
QPTR1   002002R          629#  630*  631   633*  680#  778*
QPTR2   002004R          642   643*  644   646*  681#  779*
RAND$  = 104417          368#
RANNUM  000054R          337#
RCVSW   002100R          430   580   590*  613*  710#
RERCK   010646R          479   510   789#  806
RESTRT  001204R          356   551#
RES1    000056R          339#
RES2    000060R          340#
RINT    001266R          570#  727   733   739   745   751   757   763   769
RSTRT   000112R          356#
SBADR   000102R          349#
SETCSR  000346R          390   398#
SETQTS  010600R          388   411   436   551   778#
SETUP   000442R          413   420#
SOFCNT  000042R          326#
SOFERS= 104406           368#
SOFPAS  000046R          334#
SPDINT  000032R          328#
SPSIZ  = 000040            1   361
SR1     000016R          321#
SR2     000020R          322#
SR3     000022R          323#
```



```
SR4     000024R          324#
START   000224R          327   371#
STAT    000026R          326#
STATRL  002034R          535   695#  804
STATUS  002012P          525*  684#  695   794*  796*  797*
STRTUP  000612R          444   455#
SVR0    000062R          341#
SVR1    000064R          342#
SVR2    000066R          343#
SVR3    000070R          344#
SVR4    000072R          345#
SVR5    000074R          346#
SVR6    000076R          347#
SYSCNT  000052R          336#
TIMR    000654R          472#  492
TINT    001546R          629#  730   736   742   748   754   760   766   772
TMOUT   000756R          478   501#
TO      002140R          631   633   644   646   712#  778   779
TRPDFD= 000022           368#
TSERV   001606R          536   642#
VECTOR  000010R          317#  373
WASADR  000104R          351#
WDFR    000116R          358#
WDTO    000114R          357#  372*  378*
XFLAG   000005R          315#
XRAD0   002040R          421   556   614   656   698#
XRAD1   002042R          699#
XRAD2   002044R          700#
XRAD3   002046R          701#
XRAD4   002050R          702#
XRAD5   002052R          703#
XRAD6   002054R          704#
XRAD7   002056R          705#
XRBP0   002014R          421*  422*  556*  557*  591*  592*  614*  652*  653   656*  657*  686#
XRBP1   002016R          687#
XRBP2   002020R          688#
XRBP3   002022R          689#
XRBP4   002024R          690#
XRBP5   002026R          691#
XRBP6   002030R          692#
XRBP7   002032R          693#
XRB0    002320R          426   595   698   715#  799
XRB1    003126R          699   716#
XRB2    003734R          700   717#
XRB3    004542R          701   718#
XRB4    005350R          702   719#
XRB5    006156R          703   720#
XRB6    006764R          704   721#
XRB7    007572R          705   722#
XSTRT   000704R          481#  563
.      = 010760R          709#  710#  711#  712#  713#  715#  716#  717#  718#  719#  720#  721#  722#

. ABS.  000000     000
        010760     001
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

XVTAB0,XVTAB0/SOL/CRF:SYM=DDXCOM,XVTAB0
RUN-TIME: 1 2 .3 SECONDS
RUN-TIME RATIO: 17/4=3.8
CORE USED:  7K  (13 PAGES)
```