.REM _

IDENTIFICATION

PRODUCT CODE: AC-E706G-MC

PRODUCT NAME: CXDCAG0 DC11 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT:

DCA IS AN IOMOD THAT EXERCISES UP TO SIXTEEN DC11 ASYNCHRONOUS INTERFACES. IT IS CAPABLE OF EXERCISING ALL DC11 MODELS . IT USES MAINTENANCE MODE TO XMIT AND RECEIVE A BINARY COUNT PATTERN OUTPUT AND RECEIVED IN 64. CHARACTER BURSTS. THE MAJOR PORTION OF THE ERROR CHECKING IS DEFERRED TO LEVEL 0. ALL LINES SELECTED FOR TEST (UP TO 16 DC11'S WITH CONTIGUOUS ADDRESSES AND VECTORS) ARE ACTIVATED AND RUN CONCURRENTLY. ALL TRANSMIT AND RECEIVE ERRORS ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS:

HARDWARE: AT LEAST ONE DC11 INTERFACE

STORAGE:: DCA REQUIRES:
1. DECIMAL WORDS: 868
2. OCTAL WORDS: 1544
3. OCTAL BYTES: 3310

3. PASS DEFINITION:

ONE PASS OF THE DCA MODULE CONSISTS OF TRANSMITTING AND RECEIVING 8192. (TOTAL) CHARACTERS.

4. EXECUTION TIME:

VARIES WITH BAUD RATE BUT SHOULD TAKE AN AVERAGE OF ONE MINUTES TO COMPLETE ONE PASS WHEN RUNNING ALONE.

5. CONFIGURATION PARAMETERS:

DEFAULT PARAMETERS:

DVA: 174000, VCT: 300, BR1: 5, BR2: 0, DVC: 1

REQUIRED PARAMETERS:

AT CONFIGURATION TIME THE USER MUST SPECIFY:

VCT: VECTOR ADDRESS OF FIRST DC11 IF NOT 300
DVC: NO OF DC11'S IF GREATER THAN 1

6. DEVICE OPTION SETUP:

NONE REQUIRED

7. MODULE OPERATION:

7.1 TEST SEQUENCE:

A. START: USING THE DEVICE SELECTION PARAMETER "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. STUP: THIS ROUTINE INITIALIZES ALL TABLES, BUFFERS, FLAGS AND COUNTERS, THEN PROCEEDS TO TURN ON THE INTERRUPTS FOR ALL ACTIVE LINES. IT USES THE CONTENTS OF THE ACTIVE DEVICE TABLE TO FIND OUT WHICH LINES TO KICK OFF. AFTER INITIALIZING ALL LINES IT WAITS FOR COMPLETION OF 64 TRANSMITTER AND RECEIVER INTERRUPTS VIA A BREAK LOOP. IF THE 64 INTERRUPTS HAVE OCCURRED ON BOTH TRANSMITTER AND RECEIVER, OR IF THE BREAK LOOP TIMES OUT, CONTROL PASSES TO ERRCHK.

(7.1 CONT'D)

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

E. TSERV: THIS CODE RETRIEVES A POINTER FROM THE FIFO QUEUE AND BUILDS THE CSR ADDRESS. THE FOLLOWING SEQUENCE IS EXECUTED:

1. TEST FOR END OF 64. CHAR BURST - IF END EXIT - IF NOT GO TO 2

2. TEST READY FLAG - IF NOT ASSERTED GO REPORT FALSE INTERRUPT - IF ASSERTED PROCEED TO STEP 3

3. COUNT THE INTERRUPT FOR INDIVIDUAL LINE

4. GENERATE AND OUTPUT NEXT CHARACTER, KEEP TRACK OF THE NUMBER OF CHARACTERS OUTPUT ON THE LINE, AND THEN EXIT BACK TO THE MONITOR.

F. RINT: THE RECEIVER SERVICE ROUTINE STORES DATA AND STATUS INFORMATION IN A RECEIVER STARTUP TABLE, TESTS FOR THE END OF A 64. CHAR XFR SEQUENCE AND THEN EXECUTES AN "RTI" . IT ALSO COUNTS RECEIVE INTERRUPTS IN A SEPARATE COUNTER FOR EACH LINE. SEPARATE COUNTER.

G. ERRCHK: THE BULK OF THE ERROR CHECKING AND REPORTING IS DONE HERE AT THE END OF EACH 64. CHAR. BURST. THE FOLLOWING SEQUENCE IS EXECUTED:

1. TURN OFF RCVR AND XMTR INTR. ENABLES FOR ALL ACTIVE LINES

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

(7.1, SECTION G CONT'D)

A.) PARITY, FRAMING AND OVER-RUN ERRORS.
B.) RCVR FALSE INTERRUPTS
C.) DATA COMPARE ERRORS. ONLY IF A AND B DID NOT OCCUR.

3. CHECK RECEIVER AND TRANSMITTER INTERRUPT COUNTS FOR EACH LINE TO BE SURE THAT NO LINES WERE DROPPED OR HAD TOO MANY INTERRUPTS.

4. GO TO THE ENPS ROUTINE AFTER CHECKING ALL 64 ENTRIES.

H. ENPS: COUNT THE 64. CHAR BURST AND TEST FOR 128. BURSTS (8192 CHARS). IF NOT END OF PASS GO TO I. IF END REPORT END OF PASS AND GO TO C.

I. RESYNC: RESYNC THE DATA BUFFERS AND THEN RESTART AT STEP C.

7.2 DESCRIPTION OF TABLES, QUEUES, AND BUFFERS

A. RSTAB: THIS IS A 128. WORD STATUS TABLE CONSISTING OF 64. TWO WORD ENTRIES. IT GETS LOADED DURING RECEIVER INTERRUPT SERVICE AND CHECKED AT THE END OF EACH 64. CHAR BURST. EACH ENTRY HAS THE FOLLOWING FORMAT:

1ST WORD: CONTENTS OF RCSR

2ND WORD: LO BYTE = RCVD DATA BYTE
HI BYTE = LINE NUMBER

B. RCNT: 16 BYTE TABLE CONTAINING AN 8 BIT INTERRUPT COUNTER FOR EACH RCVR. THE APPROPRIATE BYTE GETS INCREMENTED DURING RCVR INTR SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.

C. TCNT: 16 BYTE TABLE CONTAINING AN 8-BIT INTERRUPT COUNTER FOR EACH TRANSMITTER. THE APPROPRIATE BYTE GETS INCREMENTED DURING DEFERRED INTR. SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.

(7.2 CONT'D)

D. DCNT: 16 BYTE TABLE CONTAINING AN 8-BIT DATA COUNTER FOR EACH LINE. THE APPROPRIATE BYTE GETS INCREMENTED EACH TIME A CHARACTER IS TRANSMITTED ON THE LINE, AND CLEARED BEFORE THE BEGINNING OF EACH 64. WORD BURST.

E. TQ: 16 WORD FIFO QUEUE FOR TRANSMITTER SERVICE. LOADED DURING XMTR INTERRUPT SERVICE WITH A POINTER TO THE CSR ADDRESS AND UNLOADED DURING DEFERRED XMTR SERVICE.

F. XBUF: 16 BYTE XMTP DATA BUFFERS - ONE BYTE/XMTR

G. RBUF: 16 BYTE RCVR DATA BUFFERS - ONE BYTE/RCVR.

H. JSRTAB: A 128 WORD TABLE THAT CONTAINS 64 JSR INSTRUCTIONS WITH TWO TRAILING ARGUMENTS. EACH RECEIVER AND EACH XMTR HAS AN ASSIGNED JSR IN THE TABLE OF THE FOLLOWING FORMAT:

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

WHERE THE 0 GETS OVERLAYED WITH THE ADDRESS OF THE CSR FOR LINE N AND N IS THE LINE NO. IN OCTAL (00-17)

8. OPERATOR OPTIONS:

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

B. THE USER CAN MODIFY "DVID1" (APC 14) TO SELECT OR DESELECT INDIVIDUAL DC11'S.

9. NON-STANDARD PRINTOUTS:

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

```
                ;
                ;DC11 A-E DEC/X11 EXERCISER MODULE

000000'                 IDMOD   <DCAG >,174000,300,5,,,128.,,24
000000'                 MODULE  140000,DCAG ,174000,300,5,,,128.,,24
                        .TITLE  DCAG DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 041504  043501     040  MODNAM: .ASCII  /DCAG / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 174000                  ADDR:   174000+0                ;1ST DEVICE ADDR.
000010' 000300                  VECTOR: 300+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' 000200                  ICONT:  128.                    ;# OF ITERATIONS PER PASS=128.
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                  CSRAS:  OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SMADR:                          ;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' 000434'                 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' 000024                  IDNUM:  24                      ;MODULE IDENTIFICATION NUMBER=24
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR

000224'                         MODSP:
                                ;*****************************************************************************
  358                           ;THIS ROUTINE SETS UP THE VECTORS FOR ALL SELECTED LINES TO POINT
  359                           ;TO THE APPROPRIATE JSR IN THE JSR LINK TABLE
  360 000224' 012767  000040  177662  START:  MOV     #32.,WDTO       ;32 WORDS TO MEM PER ITERATION
  361 000232' 012767  000040  177656          MOV     #32.,WDFR       ;32 WORDS TO MEM
  362 000240' 012767  000200  177652          MOV     #128.,INTR      ;128. INTERRUPTS
  363 000246' 016700  177536                  MOV     VECTOR,R0       ;SET R0 TO POINT TO THE 1ST VECTOR
  364 000252' 016701  177536                  MOV     DVID1,R1        ;LOAD R1 WITH DEVICE SELECTION PARAMETER
  365 000256' 010167  002424                  MOV     R1,DVICE        ;INITIALIZE TO RECORD ANY LINES DROPPED
  366 000262' 012702  002710'                 MOV     #JSRTAB,R2      ;SET UP R2 TO POINT TO JSR TABLE
  367 000266' 006201                  1S:     ASR     R1              ;SHIFT SELECT BIT INTO "C"
  368 000270' 103020                          BCC     3S              ;BR IF NOT SELECTED
  369 000272' 010220                          MOV     R2,(R0)+        ;SET UP RCVR INTR POINTER
  370 000274' 116720  177512                  MOVB    BR1,(R0)+       ;SET UP RCVR PRIORITY LEVEL
  371 000300' 105720                          TSTB    (R0)+           ;MOVE POINTER
  372 000302' 062702  000010                  ADD     #10,R2          ;POINT R2 TO XMTR ENTRY IN JSR TABLE
  373 000306' 010220                          MOV     R2,(R0)+        ;SET UP XMTR INTR POINTER
  374 000310' 116720  177476                  MOVB    BR1,(R0)+       ;SET UP XMTR PRIORITY LEVEL
  375 000314' 105720                          TSTB    (R0)+           ;MOVE POINTER
  376 000316' 062702  000010                  ADD     #10,R2          ;POINT R2 TO RCVR ENTRY FOR NEXT LINE
  377 000322' 022702  003310'         2S:     CMP     #JSRTAB+400,R2  ;IS THE POINTER AT THE END OF THE TABLE?
  378 000326' 001357                          BNE     1S              ;BR IF NOT
  379 000330' 000405                          BR      SETCSR          ;GO SET UP CSR ADDRESSES
  380 000332' 062700  000010          3S:     ADD     #10,R0          ;UPDATE VECTOR POINTER
  381 000336' 062702  000020                  ADD     #20,R2          ;UPDATE JSR TABLE POINTER
  382 000342' 000767                          BR      2S              ;GO CHECK FOR END OF TABLE
  383
  384                           ;THIS ROUTINE SETS UP THE JSR TABLE SUCH THAT THE APPROPRIATE
  385                           ;CSR ADDRESS IS INCLUDED AS THE 3RD WORD OF EACH ENTRY
  386 000344' 016700  177436  SETCSR: MOV     ADDR,R0         ;GET THE FIRST CSR ADDRESS INTO R0
  387 000350' 016701  177440          MOV     DVID1,R1        ;LOAD R1 WITH THE DEVICE SELECTION PARAMETER
  388 000354' 001002                  BNE     1S              ;NO BRANCH IF DVID1 = 0
  389 000356' 104410  000000'         ENDS,BEGIN              ;
  390 000362' 012702  002714'         1S:     MOV     #JSRTAB+4,R2    ;POINT R2 TO CSR ADDRESS ENTRY
  391 000366' 006201          2S:     ASR     R1              ;SHIFT SELECT BIT INTO "C"
  392 000370' 103014                  BCC     4S              ;BR IF LINE NOT SELECTED
  393 000372' 010012                  MOV     R0,(R2)         ;PUT RCVR CSR ADDRESS IN TABLE
  394 000374' 022020                  CMP     (R0)+,(R0)+     ;GENERATE XMTR CSR ADDRESS IN R0
  395 000376' 062702  000010          ADD     #10,R2          ;POINT TO XMTR SLOT IN JSR TABLE
  396 000402' 010012                  MOV     R0,(R2)         ;PUT XMTR CSR ADDRESS IN THE TABLE
  397 000404' 022020                  CMP     (R0)+,(R0)+     ;GENERATE NEXT RCVR CSR ADDRESS IN R0
  398 000406' 062702  000010          ADD     #10,R2          ;POINT TO RCVR SLOT IN JSR TABLE
  399 000412' 022702  003314'         3S:     CMP     #JSRTAB+404,R2  ;IS POINTER BEYOND END OF TABLE?
  400 000416' 001361                  BNE     2S              ;BR IF NOT
  401 000420' 000405                  BR      RESTRT          ;GO SET UP ACTIVE DEVICE TABLE.
  402 000422' 062700  000010          4S:     ADD     #10,R0          ;UPDATE CSR ADDRESS
  403 000426' 062702  000020                  ADD     #20,R2          ;UPDATE JSR TABLE POINTER
```

```
                                        BR      3$              ;GO TEST FOR END OF TABLE

                        ;THIS ROUTINE CLEARS BUFFERS AND TABLES, INITIALIZES FLAGS, AND STARTS
                        ;UP ALL SELECTED LINES.
000434' 004767 002122   RESTRT: JSR     PC,CLRBUF       ;GO CLEAR XMTR. AND RCVR. BUFFERS
000440' 004767 002152   STUP1:  JSR     PC,DTAB         ;SET UP THE ACTIVE DEVICE TABLE.
000444' 004767 002130           JSR     PC,CLRTAB       ;GO CLEAR TABLES AND QUEUES
000450' 005067 002072           CLR     TXCNT           ;CLEAR TX TOTAL INTERRUPT COUNTER.
000454' 005067 002070           CLR     RXCNT           ;CLEAR RX TOTAL INTERRUPT COUNTER.
000460' 012767 001764' 002066   MOV     #RSTAB,SVPTR    ;INITIALIZE RCVR. STATUS TABLE POINTER
000466' 012767 002444' 002062   MOV     #TQ,QPTR1       ;SET UP XMTR FIFO QUEUE POINTERS
000474' 012767 002444' 002056   MOV     #TQ,QPTR2
000502' 016700 002176           MOV     ACTDEV,R0       ;GET COUNT OF ACTIVE DEVICES
000506' 116002 002664'  1$:     MOVB    DEVTAB(R0),R2   ;GET AN ACTIVE LINE NO.
000512' 004767 001102           JSR     PC,GETADR       ;GO BUILD CSR ADDRESS IN R3
000516' 005763 000002           TST     2(R3)           ;READ RCV DBR TO FLUSH DONE BIT
000522' 052713 003130           BIS     #3130,(R3)      ;ENABLE RECEIVER INTERRUPTS
000526' 052763 000434  000004   BIS     #434,4(R3)      ;ENABLE MAINT. MODE
000534' 105262 002504'          INCB    XBUF(R2)        ;
000540' 116263 002504' 000006   MOVB    XBUF(R2),6(R3)  ;OUTPUT CHAR ONTO TX.
000546' 105267 001774           INCB    TXCNT           ;UP COUNT OF CHARS OUTPUT.
000552' 105262 002424'          INCB    DCNT(R2)        ;COUNT CHARACTERS OUTPUT ON THIS LINE
000556' 052763 000100  000004   BIS     #100,4(R3)      ;ENABLE TX INTERRUPTS.
000564' 005300                  DEC     R0              ;COUNT ONE KICKED OFF
000566' 100347                  BPL     1$              ;GO TEST FOR NEXT ONE
000570' 012767 000006  001754   MOV     #6,CNTR         ;INITIALIZE COUNTER TO WAIT AT LEAST
                                                        ;30 SECONDS BEFORE TIMING OUT
000576' 005004          10$:    CLR     R4
000600'                 2$:
000600' 104407 000000'          BREAK$,REGIN            ;TEMPORARY RETURN TO MONITOR....
000604' 104407 000000'          BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000610' 122767 000100  001731   CMPB    #64.,TXCNT+1    ;64 TRANSMITTER INTERRUPTS?
000616' 003004                  BGT     3$              ;NO- BRANCH TO WAIT
000620' 022767 000100  001722   CMP     #64.,RXCNT      ;YES- 64 RECEIVER INTERRUPTS?
000626' 003405                  BLE     4$              ;YES- GO CHECK FOR ERRORS
000630' 005304          3$:     DEC     R4              ;TIMEOUT?
000632' 001362                  BNE     2$              ;NO- WAIT SOME MORE
000634' 005367 001712           DEC     CNTR            ;EACH PASS OF THE SMALL LOOP TAKES
                                                        ;AT LEAST 5 SECONDS
000640' 001356                  BNE     10$             ;BRANCH IF NOT DONE WITH 6 PASSES OF
                                                        ;THE SMALL COUNTER
000642' 000167  000272  4$:     JMP ERRCHK

                        ;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
000646' 010577 001704   TINT:   MOV     R5,@QPTR1       ;STORE CONTENTS OF R5 IN THE QUEUE
000652' 062767 000002  001676   ADD     #2,QPTR1        ;UPDATE THE QUEUE POINTER
000660' 022767 002504' 001670   CMP     #TQ+40,QPTR1    ;POINTER AT END OF QUEUE?
000666' 001003                  BNE     1$              ;BR IF NOT
000670' 012767 002444' 001660   MOV     #TQ,QPTR1       ;RESET THE POINTER
000676' 012605          1$:     MOV     (R6)+,R5        ;RESTORE THE OTHER GUY'S R5
                                ;-----------------------------------------------------------------
000700' 000004  000000' 000706'         PIRQ$,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
000706' 017700 001646   TSERV:  MOV     @QPTR2,R0       ;GET POINTER FROM THE QUEUE
000712' 062767 000002  001640   ADD     #2,QPTR2        ;UPDATE THE QUEUE POINTER
000720' 022767 002504' 001632   CMP     #TQ+40,QPTR2    ;POINTER AT HIGH LIMIT
000726' 001003                  BNE     1$              ;BR IF NOT
000730' 012767 002444' 001622   MOV     #TQ,QPTR2       ;RESET THE POINTER
000736' 012001          1$:     MOV     (R0)+,R1        ;MOV CSR ADDRESS INTO R1
000740' 011000                  MOV     (R0),R0         ;MOV LINE # INTO R0
000742' 105267 001601           INCB    TXCNT+1         ;COUNT TOTAL TX INTERRUPTS.
000746' 105260 002404'          INCB    TCNT(R0)        ;COUNT THE INTERRUPT
000752' 105711                  TSTB    (R1)            ;XMTR READY FLAG ASSERTED?
000754' 100017                  BPL     4$              ;BR IF NOT
000756' 122767 000100  001562   CMPB    #64.,TXCNT      ;64 CHARACTERS TRANSMITTED?
000764' 001427                  BEQ     5$              ;YES- BRANCH TO EXIT
000766' 105260 002504'          INCB    XBUF(R0)        ;GENERATE NEXT DATA BYTE
000772' 116061 002504' 000002   MOVB    XBUF(R0),2(R1)  ;LOAD THE XMTR BUFFER
001000' 105267 001542           INCB    TXCNT           ;UP TOTAL COUNT OF CHARS OUTPUT.
001004' 105260 002424'          INCB    DCNT(R0)        ;COUNT CHARACTERS OUTPUT ON THIS LINE
001010' 104400 000000'          EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
001014' 010167 177060   4$:     MOV     R1,CSRA         ;SAVE CSR ADDRESS
001020' 011167 177056           MOV     (R1),ACSR       ;SAVE CONTENTS OF THE CSR
001024' 142711 000100           BICB    #100,(R1)       ;DISABLE XMTTR INTERRUPT
001030' 012767 000011  177050   MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                ;*****************************************************************
001036' 104405  000000' 000000  ERROR$,BEGIN,NULL       ;XMTTR FALSE INTERRUPT
                                ;*****************************************************************
001044'
001044' 104400  000000' 5$:
                        EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                        ;RECEIVER INTERRUPT SERVICE-ENTERED VIA APPROPRIATE JSR TABLE ENTRY
                        ;STORES PERTINENT INFORMATION IN THE RECEIVER STATUS TABLE THAT WILL
                        ;BE CHECKED IF 64. CHARACTERS HAVE BEEN RECEIVED
001050' 010046  RINT:   MOV     R0,-(R6)        ;SAVE R0 AND R1 ON THE STACK
001052' 010146          MOV     R1,-(R6)
001054' 022767 000100  001466   CMP     #64.,RXCNT      ;64 CHAR'S RECEIVED?
001062' 003414                  BLE     1$              ;YES- BRANCH
001064' 016700 001464           MOV     SVPTR,R0        ;NO- GET RCVR STATUS TABLE POINTER
001070' 011501                  MOV     (R5),R1         ;GET RCVR CSR ADDRESS
001072' 011120                  MOV     (R1),(R0)+      ;SAVE THE RCSR CONTENTS
001074' 116120 000002           MOVB    2(R1),(R0)+     ;SAVE THE RCVD DATA
001100' 051160 177775           BIS     (R1),-3(R0)     ;REREAD CSR IN CASE ERROR SET BETWEEN THE
                                                        ;LAST TWO INSTRUCTIONS
001104' 116520 000002           MOVB    2(R5),(R0)+     ;SAVE THE LINE NUMBER
001110' 010067 001440           MOV     R0,SVPTR        ;SAVE THE UPDATED STATUS TABLE POINTER
001114' 016505 000002   1S:     MOV     2(R5),R5        ;GET LINE NO. INTO R5
001120' 105265 002364'          INCB    RCNT(R5)        ;COUNT THE INTERRUPT FROM THIS LINE
001124' 005267 001420           INC     RXCNT           ;INCREMENT RX INTERRUPT TOTAL COUNT.
001130' 012601          2S:     MOV     (R6)+,R1        ;RESTORE THE OTHER GUY'S REGISTER
001132' 012600                  MOV     (R6)+,R0
001134' 012605                  MOV     (R6)+,R5
001136' 000002                  RTI                     ;RETURN CONTROL BACK TO OTHER GUY

                        ;RECEIVER ERROR CHECKING AND CLEANUP ROUTINES
```

```
                                ;THIS ROUTINE DISABLES INTERRUPTS FROM ALL ACTIVE LINES
001140' 016700  001540          ERRCHK: MOV     ACTDEV,R0       ;GET COUNT OF ACTIVE DEVICES
001144' 116002  002664'         1$:     MOVB    DEVTAB(R0),R2   ;GET ACTIVE LINE NO.
001150' 004767  000444                  JSR     PC,GETADR       ;GO BUILD ADDRESS IN R3
001154' 042713  000100                  BIC     #100,(R3)       ;TURN OFF RECEIVER.
001160' 042763  000100  000004          BIC     #100,4(R3)      ;TURN OFF TRANSMITTER.
001166' 005300                          DEC     R0              ;COUNT ONE GUY OFF
001170' 100365                          BPL     1$              ;BR TIL ALL OFF

                                ;THIS ROUTINE SCANS THROUGH THE 64 ENTRY RECEIVER STATUS TABLE
                                ;CHECKING FOR AND REPORTING ANY ERRORS.
001172' 026727  001352  000100  CHK1:   CMP     RXCNT,#64.      ;MAKE SURE COUNT IS NO LARGER THAN TABLE
001200' 003403                          BLE     1$
001202' 012767  000100  001340          MOV     #64.,RXCNT
001210' 012701  001764'         1$:     MOV     #RSTAB,R1       ;GET STATUS TABLE POINTER
001214' 005000                  2$:     CLR     R0              ;INDICATE NO HARDWARE FAILURES YET.
001216' 116102  000003                  MOVB    3(R1),R2        ;GET LINE NO. INTO R2
001222' 005711                          TST     (R1)            ;ERROR SET?
001224' 100027                          BPL     4$              ;BRANCH IF NOT
001226' 004767  000456                  JSR     PC,RCVERR       ;GO SETUP TO REPORT ERROR
001232' 005067  176650                  CLR     ERRTYP          ;UNKNOWN ERROR
                                        ;*****************************************************************
001236' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CARRIER TRANS + RING + OVERRUN
                                        ;*****************************************************************
001244' 032711  020000                  BIT     #BIT13,(R1)     ;RING INDICATOR SET?
001250' 001415                          BEQ     4$              ;BRANCH IF NOT
001252' 104403  000000' 001450'         MSGN$,BEGIN,RING        ;ASCII MESSAGE CALL WITH COMMON HEADER
001260' 012701  000001                  MOV     #1,R1           ;SETUP TO DROP LINE
001264' 000401                          BR      .+4
001266' 006301                  3$:     ASL     R1              ;SHIFT BIT TO ALLIGN WITH INDICATOR IN DVICE
001270' 005302                          DEC     R2
001272' 100375                          BPL     3$
001274' 040167  001406                  BIC     R1,DVICE        ;DROP THE LINE
001300' 000167  000424                  JMP     ENPS            ;SKIP REST OF CHECKING SINCE RING INDICATOR
                                                                ;SET WILL CAUSE ALL COUNTS TO BE BAD
001304' 105711                  4$:     TSTB    (R1)            ;POINT TO LO BYTE OF CSR
001306' 100410                          BMI     5$              ;BR IF DONE WAS SET
001310' 004767  000374                  JSR     PC,RCVERR       ;SETUP FOR ERROR REPORT
001314' 012767  000011  176564          MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;*****************************************************************
001322' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;RECEIVER FALSE INTERRUPT
                                        ;*****************************************************************
001330' 105262  002524'         5$:     INCB    RBUF(R2)        ;BUMP EXPECTED DATA
001334' 005700                          TST     R0              ;HARDWARE ERRORS?
001336' 001014                          BNE     6$              ;DO NOT REPORT DATA ERRORS THEN.
001340' 142762  000340  002524'         BICB    #340,RBUF(R2)   ;MASK OFF BITS <7:5> TO CHECK ONLY
001346' 142761  000340  000002          BICB    #340,2(R1)      ;FIVE BITS ON
001354' 126261  002524' 000002          CMPB    RBUF(R2),2(R1)  ;DID RCVD DATA CHECK OK?
001362' 001402                          BEQ     6$              ;BR IF YES
001364' 004767  000246                  JSR     PC,DATBAD       ;GO REPORT DATA ERROR
001370' 022121                  6$:     CMP     (R1)+,(R1)+     ;POINT R1 TO NEXT TABLE ENTRY
001372' 005367  001152                  DEC     RXCNT           ;ALL CHARS RECEIVED CHECKED?
001376' 001306                          BNE     2$              ;NO- GO CHECK NEXT ENTRY

                                ;THIS ROUTINE REPORTS ANY LINE RECEIVING AN INCORRECT NUMBER OF INTERRUPTS
001400' 016701  001300          CKLINS: MOV     ACTDEV,R1       ;GET ACTIVE DEVICE COUNT
```



```
001404' 116102  002664'         3$:     MOVB    DEVTAB(R1),R2   ;GET ACTIVE DEVICE LINE NO.
001410' 126262  002364' 002424'         CMPB    RCNT(R2),DCNT(R2)       ;CORRECT NUMBER OF RECVR INTERRUPTS?
001416' 001402                          BEQ     4$              ;BR IF YES
001420' 004767  000066                  JSR     PC,BADR         ;GO REPORT BAD RCVR
001424' 126262  002404' 002424' 4$:     CMPB    TCNT(R2),DCNT(R2)       ;CORRECT NUMBER OF XMTR INTERRUPTS?
001432' 001402                          BEQ     5$              ;BR IF YES
001434' 004767  000114                  JSR     PC,BADT         ;GO REPORT BAD XMTR
001440' 005301                  5$:     DEC     R1              ;COUNT ONE GUY CHECKED
001442' 100360                          BPL     3$              ;BR TIL ALL CHECKED
001444' 000167  000260                  JMP     ENPS            ;GO CHECK FOR END OF PASS


001450' 001454'                 RING:   MRING
001452' 177777                          -1

001454' 051045  047111  020107  MRING:  .ASCIZ  /%RING SET- BAD LINE DROPPED%/
001462' 042523  026524  041040
001470' 042101  046040  047111
001476' 020105  051104  050117
001504' 042520  022504      000
            001512'             .EVEN

                                ;ROUTINE TO REPORT BAD LINES (TOO MANY OR TOO FEW INTERRUPTS)
001512' 004767  000102  BADR:   JSR     PC,GETADR       ;GO BUILD CSR ADDRESS
001516' 010367  176356          MOV     R3,CSRA         ;SAVE CSR ADDRESS
001522' 116267  002424' 176352  MOVB    DCNT(R2),ACSR   ;CHARACTERS XMTD
001530' 116267  002364' 176346  MOVB    RCNT(R2),ASTAT  ;# OF RCVR INTERRUPTS

001536' 012767  000014  176342          MOV     #14,ERRTYP      ;WRONG # OF INTERRUPTS
                                        ;*****************************************************************
001544' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT NUMBER OF RCVR INTERRUPTS
                                        ;*****************************************************************
                                        ;NOTE THAT CSRC VALUE IS # OF CHARACTERS
                                        ;TRANSMITTED, AND STATC VALUE IS # OF
                                        ;RECEIVER INTERRUPTS

001552' 000207                          RTS     PC              ;RETURN TO CALLER

001554' 004767  000040  BADT:   JSR     PC,GETADR       ;GO BUILD CSR ADDRESS
001560' 022323                  CMP     (R3)+,(R3)+     ;MAKE IT A XMTR CSR ADDRESS
001562' 010367  176312          MOV     R3,CSRA         ;SAVE CSR ADDRESS
001566' 116267  002424' 176306  MOVB    DCNT(R2),ACSR   ;CHARACTERS XMITTED
001574' 116267  002404' 176302  MOVB    TCNT(R2),ASTAT  ;# OF XMTR INTERRUPTS

001602' 012767  000014  176276          MOV     #14,ERRTYP      ;WRONG # OF INTERRUPTS
                                        ;*****************************************************************
001610' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT NUMBER OF XMTR INTERRUPTS
                                        ;*****************************************************************
                                        ;NOTE THAT CSRC VALUE IS # OF CHARACTERS
                                        ;TRANSMITTED, AND STATC VALUE IS # OF
                                        ;TRANSMITTER INTERRUPTS

001616' 000207                          RTS     PC              ;RETURN TO CALLER

001620' 010203                  GETADR: MOV     R2,R3           ;GET LINE NO.
001622' 006303                          ASL     R3              ;BUILD CSR ADDRESS
```

```
001624' 006303                  ASL     R3
001626' 006303                  ASL     R3
001630' 066703  176152          ADD     ADDR,R3
001634' 000207                  RTS     PC              ;RETURN TO CALLER

                        ;ROUTINE TO REPORT RCVR DATA COMPARE ERRORS
001636' 004767  177756  DATBAD: JSR     PC,GETADR       ;GO BUILD CSR ADDRESS
001642' 010367  176232          MOV     R3,CSRA         ;SAVE CSR ADDRESS
001646' 116167  000002  176234  MOVB    2(R1),AWAS      ;SAVE BAD DATA
001654' 005721                  TST     (R1)+           ;GENERATE RCVR DATA ADDRESS
001656' 010167  176222          MOV     R1,WASADR       ;SAVE ADDRESS OF BAD DATA
001662' 005741                  TST     -(R1)           ;RESET R1
001664' 012705  002524'         MOV     #RBUF,R5        ;GENERATE ADDRESS OF GOOD DATA
001670' 060205                  ADD     R2,R5
001672' 111567  176210          MOVB    (R5),ASB        ;SAVE GOOD DATA
001676' 010567  176200          MOV     R5,SBADR        ;SAVE ADDRESS OF GOOD DATA
                                ;*****************************************************************
001702' 104404  000000'         DATERR,BEGIN            ;DATA ERROR!!!
                                ;*****************************************************************
001706' 000207                  RTS     PC              ;RETURN TO CALLER

                        ;ROUTINE TO SETUP FOR RECEIVER ERROR PRINTOUTS
001710' 005200  RCVERR: INC     R0              ;INDICATE HARDWARE ERROR.
001712' 004767  177702          JSR     PC,GETADR       ;GO BUILD CSR ADDRESS
001716' 010367  176156          MOV     R3,CSRA         ;STUFF IT IN CSRA
001722' 011167  176154          MOV     (R1),ACSR       ;GET CONTENTS IN ACSR
001726' 000207                  RTS     PC              ;RETURN TO CALLER

                        ;THIS ROUTINE CHECKS FOR AND REPORTS END OF PASS
001730'                 ENPS:
001730' 104413  000000'         ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                        ;MONITOR SHALL TEST END OF PASS
001734' 000167  176474          JMP     RESTRT

                        ;THIS ROUTINE RESTARTS EACH 64 CHAR XFR SEQUENCE
001740' 012700  002524' RESYNC: MOV     #RBUF,R0        ;RESYNC DATA FOR NEXT PASS
001744' 012701  002504'         MOV     #XBUF,R1
001750' 112021          1$:     MOVB    (R0)+,(R1)+
001752' 022700  002544'         CMP     #RBUF+20,R0     ;DONE 16 BYTES?
001756' 001374                  BNE     1$              ;BR IF NOT
001760' 000167  176454          JMP     STUP1           ;RESUME.

                        ;TABLES AND BUFFERS

001764' 000200          RSTAB:  .BLKW   128.            ;128 WORD(64 ENTRIES)RCVR STATUS TABLE
002364' 000010          RCNT:   .BLKW   8.              ;RCVR INTERRUPT COUNTERS
002404' 000010          TCNT:   .BLKW   8.              ;XMTR INTERRUPT COUNTERS
002424' 000010          DCNT:   .BLKW   8.              ;CHARACTER COUNTERS

002444' 000020          TQ:     .BLKW   16.             ;16 WORD XMTR FIFO QUEUE

002504' 000010          XBUF:   .BLKW   8.              ;16 BYTE XMTR DATA BUFFERS
002524' 000010          RBUF:   .BLKW   8.              ;16 BYTE RCVR DATA BUFFERS

                        ;POINTERS, CONSTANTS, AND VARIABLES
```



```
002544' 000000          COUNT:  OPEN                    ;END OF PASS COUNTER
002546' 000000          TXCNT:  OPEN                    ;TX TOTAL INTERRUPTS COUNTER (HIGH BYTE).
                                                        ;TOTAL CHARACTERS TRANSMITTED (LOW BYTE)
002550' 000000          RXCNT:  OPEN                    ;RX TOTAL INTERRUPTS COUNTER.
002552' 000000          CNTR:   OPEN                    ;BREAK LOOP COUNTER
002554' 000000          SVPTR:  OPEN                    ;TEMP STORAGE FOR RSTAB POINTER
002556' 000000          QPTR1:  OPEN                    ;XMTR FIFO QUEUE POINTER - LOAD
002560' 000000          QPTR2:  OPEN                    ;XMTR FIFO QUEUE POINTER - UNLOAD


                        ;SUBROUTINE TO CLEAR DATA BUFFERS AT BEGINING OF EACH NEW PASS
002562' 012700  002504' CLRBUF: MOV     #XBUF,R0        ;SET UP R0 TO POINT TO BEGINING
002566' 005020          1$:     CLR     (R0)+           ;CLEAR A WORD
002570' 022700  002544'         CMP     #RBUF+20,R0     ;END OF RCVR BUFFER?
002574' 001374                  BNE     1$              ;BR TIL ALL CLEAR
002576' 000207                  RTS     PC              ;RETURN TO CALLER

                        ;SUBROUTINE TO CLEAR TABLES AND QUEUES
002600' 012700  001764' CLRTAB: MOV     #RSTAB,R0       ;SET UP R0 TO POINT TO BEGINING
002604' 005020          1$:     CLR     (R0)+           ;CLEAR A WORD
002606' 022700  002504'         CMP     #TQ+40,R0       ;END?
002612' 001374                  BNE     1$              ;BR IF NOT
002614' 000207                  RTS     PC              ;RETURN TO CALLER


                        ;THIS ROUTINE SETS UP AN ACTIVE DEVICE TABLE TO REMEMBER HOW MANY
                        ;AND WHICH LINES WERE ACTIVE DURING TEST - IT IS USED DURING THE
                        ;ERROR CHECKING ROUTINES FOR VARIOUS PURPOSES
002616' 005000          DTAB:   CLR     R0              ;SET UP R0 AS TOTAL LINE COUNTER
002620' 005100                  COM     R0              ;INITIALLY SET TO MINUS ONE
002622' 005001                  CLR     R1              ;SET UP R1 AS ACTIVE LINE COUNTER
002624' 005101                  COM     R1              ;INITIALLY SET TO MINUS ONE
002626' 016702  000054          MOV     DVICE,R2        ;GET DEVICE SELECTION PARAMETER
002632' 005200          1$:     INC     R0              ;COUNT ONE DEVICE
002634' 022700  000020          CMP     #20,R0          ;16 LINES CHECKED?
002640' 001003                  BNE     2$              ;BR IF NOT
002642' 010167  000036          MOV     R1,ACTDEV       ;SAVE THE COUNT OF ACTIVE LINES
002646' 000207                  RTS     PC              ;RETURN TO CALLER
002650' 006202          2$:     ASR     R2              ;SHIFT SELECT BIT INTO "C"
002652' 103367                  BCC     1$              ;BR IF NOT SELECTED
002654' 005201                  INC     R1              ;COUNT ACTIVE LINE
002656' 110061  002664'         MOVB    R0,DEVTAB(R1)   ;STORE ACTIVE LINE NO. IN THE TABLE
002662' 000763                  BR      1$              ;GO TEST NEXT LINE

002664' 000010          DEVTAB: .BLKW   8.              ;16 BYTE ACTIVE DEVICE TABLE
002704' 000000          ACTDEV: OPEN                    ;STORES COUNT OF NO. OF ACTIVE LINES MINUS ONE
002706' 000000          DVICE:  OPEN                    ;DEVICE SELECTION INDICATOR


                        ;JSR LINK TABLE CONSISTING OF 32 JSR'S (16 RCVR AND 16 XMTR) THAT
                        ;LINK THE INTERRUPTS TO THE COMMON SERVICE ROUTINES
002710' 004567  176134  JSRTAB: JSR     R5,RINT         ;RECEIVER LINK FOR LINE 0
002714' 000000                  0                       ;SET UP WITH RCVR CSR ADDRESS
002716' 000000                  0                       ;LINE NUMBER
002720' 004567  175722          JSR     R5,TINT         ;XMTR LINK FOR LINE 0
002724' 000000                  0                       ;SET UP WITH XMTR CSR ADDRESS
002726' 000000                  0                       ;LINE NUMBER
002730' 004567  176114          JSR     R5,RINT         ;LINK FOR LINE 1
```

```
  740  002734' 000000                          0
  741  002736' 000001                          1
  742  002740' 004567  175702                  JSR     R5,TINT
  743  002744' 000000                          0
  744  002746' 000001                          1
  745  002750' 004567  176074                  JSR     R5,RINT         ;LINK FOR LINE 2
  746  002754' 000000                          0
  747  002756' 000002                          2
  748  002760' 004567  175662                  JSR     R5,TINT
  749  002764' 000000                          0
  750  002766' 000002                          2
  751  002770' 004567  176054                  JSR     R5,RINT         ;LINK FOR LINE 3
  752  002774' 000000                          0
  753  002776' 000003                          3
  754  003000' 004567  175642                  JSR     R5,TINT
  755  003004' 000000                          0
  756  003006' 000003                          3
  757  003010' 004567  176034                  JSR     R5,RINT         ;LINK FOR LINE 4
  758  003014' 000000                          0
  759  003016' 000004                          4
  760  003020' 004567  175622                  JSR     R5,TINT
  761  003024' 000000                          0
  762  003026' 000004                          4
  763  003030' 004567  176014                  JSR     R5,RINT         ;LINK FOR LINE 5
  764  003034' 000000                          0
  765  003036' 000005                          5
  766  003040' 004567  175602                  JSR     R5,TINT
  767  003044' 000000                          0
  768  003046' 000005                          5
  769  003050' 004567  175774                  JSR     R5,RINT         ;LINK FOR LINE 6
  770  003054' 000000                          0
  771  003056' 000006                          6
  772  003060' 004567  175562                  JSR     R5,TINT
  773  003064' 000000                          0
  774  003066' 000006                          6
  775  003070' 004567  175754                  JSR     R5,RINT         ;LINK FOR LINE 7
  776  003074' 000000                          0
  777  003076' 000007                          7
  778  003100' 004567  175542                  JSR     R5,TINT
  779  003104' 000000                          0
  780  003106' 000007                          7
  781  003110' 004567  175734                  JSR     R5,RINT         ;LINK FOR LINE 10
  782  003114' 000000                          0
  783  003116' 000010                          10
  784  003120' 004567  175522                  JSR     R5,TINT
  785  003124' 000000                          0
  786  003126' 000010                          10
  787  003130' 004567  175714                  JSR     R5,RINT         ;LINK FOR LINE 11
  788  003134' 000000                          0
  789  003136' 000011                          11
  790  003140' 004567  175502                  JSR     R5,TINT
  791  003144' 000000                          0
  792  003146' 000011                          11
  793  003150' 004567  175674                  JSR     R5,RINT         ;LINK FOR LINE 12
  794  003154' 000000                          0
  795  003156' 000012                          12
```



```
  796  003160' 004567  175462                  JSR     R5,TINT
  797  003164' 000000                          0
  798  003166' 000012                          12
  799  003170' 004567  175654                  JSR     R5,RINT         ;LINK FOR LINE 13
  800  003174' 000000                          0
  801  003176' 000013                          13
  802  003200' 004567  175442                  JSR     R5,TINT
  803  003204' 000000                          0
  804  003206' 000013                          13
  805  003210' 004567  175634                  JSR     R5,RINT         ;LINK FOR LINE 14
  806  003214' 000000                          0
  807  003216' 000014                          14
  808  003220' 004567  175422                  JSR     R5,TINT
  809  003224' 000000                          0
  810  003226' 000014                          14
  811  003230' 004567  175614                  JSR     R5,RINT         ;LINK FOR LINE 15
  812  003234' 000000                          0
  813  003236' 000015                          15
  814  003240' 004567  175402                  JSR     R5,TINT
  815  003244' 000000                          0
  816  003246' 000015                          15
  817  003250' 004567  175574                  JSR     R5,RINT         ;LINK FOR LINE 16
  818  003254' 000000                          0
  819  003256' 000016                          16
  820  003260' 004567  175362                  JSR     R5,TINT
  821  003264' 000000                          0
  822  003266' 000016                          16
  823  003270' 004567  175554                  JSR     R5,RINT         ;LINK FOR LINE 17
  824  003274' 000000                          0
  825  003276' 000017                          17
  826  003300' 004567  175342                  JSR     R5,TINT
  827  003304' 000000                          0
  828  003306' 000017                          17
  829
  830          000001              .END
```

```
ACSR    000102R          340#    482*    597*    613*    653*
ACTDEV  002704R          416     517     571     719*    728#
ADDR    000006R          306#    386     630
ADDR22= 001000           358#
ASB     000106R          344#    642*
ASTAT   000104R          342#    598*    614*
AWAS    000110R          345#    636*
BADR    001512R          575     595#
BADT    001554R          578     610#
BEGIN   000000R          303#    389     433     434     458     480     486     489     538     542     556     602     618
                         645     658
BIT0    = 000001         358#
BIT1    = 000002         358#
BIT10   = 002000         358#
BIT11   = 004000         358#
BIT12   = 010000         358#
BIT13   = 020000         358#    540
BIT14   = 040000         358#
BIT15   = 100000         358#
BIT2    = 000004         358#
BIT3    = 000010         358#
BIT4    = 000020         358#
BIT5    = 000040         358#
BIT6    = 000100         358#
BIT7    = 000200         358#
BIT8    = 000400         358#
BIT9    = 001000         358#
BREAK$= 104407           358#    433     434
BR1     000012R          308#    370     374
BR2     000013R          309#
BTOD$ = 104421           358#
CDAT$S= 104412           358#
CHK1    001172R          527#
CKLINS  001400R          571#
CLRBUF  002562R          408     695#
CLRTAB  002600R          410     702#
CNTR    002552R          429*    441*    688#
CONFIG  000056R          328#
COUNT   002544R          684#
CSRA    000100R          339#    481*    596*    612*    635*    652*
DATRAD  001636R          565     634#
DATCK$= 104411           358#
DATER$= 104404           358#    645
DCNT    002424R          425*    479*    573     576     597     613     675#
DEVTAB  002664R          417     518     572     724*    727#
DTAB    002616R          409     711#
DVICE   002706R          365*    548*    715     729#
DVID1   000014R          310#    364     387
END1T$= 104413           358#    658
END$  = 104410           358#    389
ENPS    001730R          549     581     657#
ERRCHK  001140R          445     517#
ERRTYP  000106R          343#    484*    536*    554*    600*    616*
EXIT$ = 104400           358#    480     489
GETADR  001620R          418     519     595     610     626#    634     651
GETPA$= 104415           358#
```



```
GWBUF$= 104414           358#
HRDCNT  000044R          323#
HRDER$= 104405           358#    486     538     556     602     618
HRDPAS  000050R          325#
ICONT   000036R          320#
ICOUNT  000040R          321#
IDNUM   000122R          350#
INIT    000030R          317#
INTR    000120R          349#    362*
JSRTAB  002710R          366     377     390     399     733#
MAP22$= 104416           358#
MODNAM  000000R          304#
MODSP   000224R          318     356#
MRING   001454R          584     587#
MSGN$ = 104403           358#    542
MSGS$ = 104402           358#
MSG$  = 104401           358#
NULL    000000           305#    486     538     556     602     618
OPEN    000000           305     311     312     313     314     331     332     333     334     335     336     337     338
                         340     342     344     345     347     348     349     358#    684     685     687     688     689
                         690     691     728     729
OTOA$ = 104420           358#
PASCNT  000034R          319#
PTR0$ = 000004           358#    458
POPSP   005726           358#
POPSP2= 022626           358#
PRTY    = 000000         309     358#
PRTY0   = 000000         358#
PRTY1   = 000040         358#
PRTY2   = 000100         358#
PRTY3   = 000140         358#
PRTY4   = 000200         358#
PRTY5   = 000240         308     358#
PRTY6   = 000300         358#
PRTY7   = 000340         358#
PS      = 177776         358#
PSW     = 177776         358#
PUSH    = 005746         358#
PUSH2   = 024646         358#
QPTR1   002556R          414*    451*    452*    453     455*    690#
QPTR2   002560R          415*    463     464*    465     467*    691#
RAND$ = 104417           358#
RANNUM  000054R          327#
RBUF    002524R          558*    561*    563     640     663     666     680#    697
RCNT    002364R          507*    573     598     673#
RCVERR  001710R          535     553     650#
RESTRT  000434R          346     401     408#    660
RESYNC  001740R          663#
RES1    000056R          329#
RES2    000060R          330#
RING    001450R          542     584#
RINT    001050R          494#    733     739     745     751     757     763     769     775     781     787     793     799
                         805     811     817     823
RSTAB   001764R          413     530     672#    702
RSTRT   000112R          346#
RXCNT   002550R          412*    437     496     508*    527     529*    567*    687#
```

```
SRADR   000102R          339#    643*
SETCSR  000344R          379     386#
SOFCNT  000042R          322#
SOFERS= 104406           358#
SOFPAS  000046R          324#
SPOINT  000032R          318#
SPSTZ = 000040           111#    351
SR1     000016R          311#
SR2     000020R          312#
SR3     000022R          313#
SR4     000024R          314#
START   000224R          317     360#
STAT    000026R          316#
STUP1   000440R          409#    668
SVPTR   002554R          413*    498     505*    689#
SVR0    000062R          331#
SVR1    000064R          332#
SVR2    000066R          333#
SVR3    000070R          334#
SVR4    000072R          335#
SVR5    000074R          336#
SVR6    000076R          337#
SYSCNT  000052R          326#
TCNT    002404R          471*    576     614     674#
TINT    000646R          451#    736     742     748     754     760     766     772     778     784     790     796     802
                         808     814     820     826
TQ      002444R          414     415     453     455     465     467     677#    704
TRPDFD= 000022           358#
TSERV   000706R          458     463#
TXCNT   002546R          411*    424*    435     470*    474     478*    685#
VECTOR  000010R          307#    363
WASADR  000104R          341#    638*
WDFR    000116R          348#    361*
WDTO    000114R          347#    360*
XQUP    002504R          422*    423     476*    477     664     679#    695
XFLAG   000056R          305#
.     = 003310R          544     592#    672#    673#    674#    675#    677#    679#    680#    727#

. ABS.  000000      000
        003310      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XDCAG0,XDCAG0/SOL/CRF:SYM=DDXCOM,XDCAG0
 RUN-TIME:  1  2  .3 SECONDS
 RUN-TIME RATIO: 15/4=3.1
 CORE USED:  7K  (13 PAGES)
```