

.REM_

IDENTIFICATION

PRODUCT CODE: AC-T881A-MC
PRODUCT TITLE: CXIECA0 IEC11-B DEC/X11 MODULE
PRODUCT DATE: JANUARY 1984
DEPARTMENT: CSS, NASHUA
AUTHOR: JOHN LEVESQUE

COPYRIGHT (C) FIRST, 1980, 1984
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754
THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED
AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE
AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR
OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO
AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
EQUIPMENT CORPORATION.
DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

5 CONFIGURATION REQUIREMENTS

PARAMETERS	DESCRIPTION UNIQUE TO THIS MODULE	DEFAULT
DVA-	DEVICE ADDRESS - IEC11-A	160010
VCT-	VECTOR ADDRESS - IEC11-A	270
BR1-	BUS REQUEST LEVEL - IEC11-A	6
BR2-	BUS REQUEST LEVEL - IEC11-B	5
DVC-	DRIVE COUNT (N/A)	1
SR1-	DEVICE ADDRESS - IEC11-B	160020
SR2-	VECTOR ADDRESS - IEC11 B	274
SR3-	IEEE BUS ADDRESS - IEC11-A	35
SR4-	IEEE BUS ADDRESS - IEC11-B	36

REQUIRED PARAMETERS:

NONE

6 DEVICE / OPTION SETUP

MAKE CERTAIN THAT THE IEC11-A AND IEC11-B ARE CONNECTED AND READY. THIS ENTAILS INSTALLING A BC08S-01 CABLE FROM IEC11-A(M7985) J1 TO IEC11-B(M7915) J1.

7 MODULE OPERATION

TEST SEQUENCE:

A. TEST DATA TRANSFER WITH INTERRUPT, CHECKING THAT:

1. 'TCS' PRODUCES 'ILLMSGF'
2. INTERRUPT OCCURS WHEN SENDING LISTENER ADDRESS
3. 'LACS' IS SET AFTER LISTENER ADDRESSING AND 'GTS'
4. INTERRUPT OCCURS WHEN SENDING TALKER ADDRESS
5. 'DATAACC' DOES NOT COME UP ON 2ND DATA BYTE BEFORE 1ST DATA BYTE IS READ
6. 1ST DATA BYTE SENT = 1ST DATA BYTE RECEIVED
7. 'DATAACC' COMES UP ON 2ND DATA BYTE AFTER READING 1ST DATA BYTE
8. 2ND DATA BYTE SENT = 2ND DATA BYTE RECEIVED

B. TEST RANDOM TALKER AND LISTENER DATA, CHECKING THAT:

1. INTERRUPT OCCURS WHEN SENDING A DATA BYTE
2. DATA BYTES SENT = DATA BYTES RECEIVED
3. 'DATAACC' COMES UP WHEN SENDING A DATA BYTE WITH 'LASTBYTE'
4. 'END' COMES UP WHEN LAST BYTE SENT
5. LAST BYTE SENT = LAST BYTE RECEIVED

C. TEST WITH IEC11-B AS TALKER AND IEC11-A AS LISTENER.

1. SET SACS
2. SET SIC
3. BUILD TALKER AND LISTENER ADDRESSES.
4. START TALKING - 256 BYTES WORTH.
5. COMPARE DATA BYTES FROM BUF1 AND BUF2.

D. TEST WITH IEC11-A AS TALKER AND IEC11-B AS LISTENER.

1. SET SACS.
2. SET SIC.
3. BUILD LISTENER AND TALKER ADDRESSES.
4. DO 256 BYTE TRANSFERS.
5. COMPARE DATA BYTES FROM BUF1 AND BUF2.

8 OPERATION OPTIONS

NONE

9 NON-STANDARD PRINTOUTS

ALL ERROR PRINTOUTS CONSIST OF TWO SECTIONS. THE FIRST SECTION IS NON-STANDARD, AND CONSISTS OF AN ERROR CODE (IE. ERROR NN) THAT IS A NUMERICAL CROSS REFERENCE TO THE SAME ERROR CONDITION AND ERROR NUMBER IN THE PDP11 DIAGNOSTIC (CZIECAO). THE SECOND SECTION CONSISTS OF A STANDARD DEC/X11 HARD ERROR (HORDER) CALL, WITH A DUMP OF THE CONTENTS OF THE DEVICE REGISTERS AND THE CONTENTS OF THE DATA BUFFER (DATBUF) IN THE FOLLOWING ORDER:

CIR, SMR, IOR, VSR, DATBUF(FOR IEC11-A PORTION OF TEST)

CIR, SMR, IOR, VSR, CSR, BCR, BAR, MCR, DATBUF(FOR IEC11-B PORTION OF TEST)

10 STARTING PROCEDURE

- A. LOAD AND START THE DEC/X11 PROGRAM ACCORDING TO THE DEC/X11 USER'S MANUAL (AC-8234D MC).
- B. ENSURE THAT THE DEVICE ADDRESSES AND VECTOR ADDRESSES ARE CORRECT FOR THE IEC11-A AND IEC11-B BUS CONTROLLER.

11 LISTING

DDXCOM HEADER

.215
.216
217 000000
000000

.SBTTL DDXCOM HEADER

IOMODX <IECA >,160010,270,6,5,0,10.,0,BUFIN,512.,512.
MODULE 150000,IECA ,160010,270,6,5,0,10.,0,BUFIN,512.,512.
.TITLE IECA DEC/X11 SYSTEM EXERCISER MODULE
; DDXCOM VERSION 6.4 28-JAN-82
.LIST BIN

000000
000000 111
000003 101
000005 000
000006 160010
000010 000270
000012 300
000013 240
000014 000001
000016 000000
000020 000000
000022 000000
000024 000000

105
040

103

```
*****
BEGIN;
MODNAM; .ASCII /IECA / ;MODULE NAME.

XFLAG; .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR; 160010+0 ;1ST DEVICE ADDR.
VECTOR; 270+0 ;1ST DEVICE VECTOR.
BR1; .BYTE PRTY6+0 ;1ST BR LEVEL.
BR2; .BYTE PRTY5+0 ;2ND BR LEVEL.
DVID1; 0+1 ;DEVICE INDICATOR 1.
SR1; OPEN ;SWITCH REGISTER 1
SR2; OPEN ;SWITCH REGISTER 2
SR3; OPEN ;SWITCH REGISTER 3
SR4; OPEN ;SWITCH REGISTER 4
*****
STAT; 150000 ;STATUS WORD.
INIT; START ;MODULE START ADDR.
SPOINT; MODSP ;MODULE STACK POINTER.
PASCNT; 0 ;PASS COUNTER.
ICONT; 10. ;# OF ITERATIONS PER PASS*10.
ICOUNT; 0 ;LOC TO COUNT ITERATIONS
SOFcnt; 0 ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT; 0 ;LOC TO SAVE TOTAL HARD ERRORS
SOFPAS; 0 ;LOC TO SAVE SOFT ERRORS PER PASS
HRDPAS; 0 ;LOC TO SAVE HARD ERRORS PER PASS
SYSCNT; 0 ;# OF SYS ERRORS ACCUMULATED
RANNUM; 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
CONFIG; ;RESERVED FOR MONITOR USE
RES1; 0 ;RESERVED FOR MONITOR USE
RES2; 0 ;RESERVED FOR MONITOR USE
SVR0; OPEN ;LOC TO SAVE R0.
SVR1; OPEN ;LOC TO SAVE R1.
SVR2; OPEN ;LOC TO SAVE R2.
SVR3; OPEN ;LOC TO SAVE R3.
SVR4; OPEN ;LOC TO SAVE R4.
SVR5; OPEN ;LOC TO SAVE R5.
SVR6; OPEN ;LOC TO SAVE R6.
CSRA; OPEN ;ADDR OF CURRENT CSR.
SBADR; ;ADDR OF GOOD DATA, OR
ACSR; OPEN ;CONTENTS OF CSR.
WASADR; ;ADDR OF BAD DATA, OR
ASTAT; OPEN ;STATUS REG CONTENTS.
ERRTYP; ;TYPE OF ERROR
ASB; OPEN ;EXPECTED DATA.
AWAS; OPEN ;ACTUAL DATA.
RSTRT; RSTRT ;RESTART ADDRESS AFTER END OF PASS
WDT0; OPEN ;WORDS TO MEMORY PER ITERATION
WDFR; OPEN ;WORDS FROM MEMORY PER ITERATION
INTR; OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM; 0 ;MODULE IDENTIFICATION NUMBER*0
```

000026 150000
000030 000252
000032 000252
000034 000000
000036 000012
000040 000000
000042 000000
000044 000000
000046 000000
000050 000000
000052 000000
000054 000000
000056
000056 000000
000060 000000
000062 000000
000064 000000
000066 000000
000070 000000
000072 000000
000074 000000
000076 000000
000100 000000
000102
000102 000000
000104
000104 000000
000106
000106 000000
000110 000000
000112 000252
000114 000000
000116 000000
000120 000000
000122 000000

ODXCOM HEADER

000124 005060'
 000126 000000
 000130 000000
 000132 001000
 000134 000000
 000136 000000
 000140 001000
 000142 000000
 000144 000000
 000146 000000
 000150 000000
 000040

RBUFVA: BUFIN
 RBUFPA: OPEN
 RBUFEA: OPEN
 RBUFSZ: 512.
 WBUFPA: OFEN
 WBUFEA: OPEN
 WBUFRQ: 512.
 WBUFSZ: OPEN
 CDERCT: OPEN
 CDWDCT: OPEN
 FREE: OPEN
 .REPT SPSIZ
 .NLIST
 .WORD 0
 .LIST
 .ENDR

;READ BUFFER VIRTUAL ADDRESS
 ;READ BUFFER PHYSICAL ADDRESS
 ;READ BUFFER EA BITS
 ;SIZE OF THE READ BUFFER
 ;WRITE BUFFER PHYSICAL ADDRESS
 ;WRITE BUFFER EA BITS
 ;WRITE BUFFER SIZE REQUESTED
 ;WRITE BUFFER SIZE AVAILABLE
 ;CDATA/DATCK ERROR COUNT
 ;CDATA/DATCK WORD COUNT
 ;RESERVED FOR FUTURE USE
 ;MODULE STACK STARTS HERE,

000252

MODSP:
 ;*****

MACRO DEFINITIONS.

```

220      .SBTTL  MACRO DEFINITIONS.
221
222      .MACRO  SET.SACS
223      BIS    @SACS,CIR(R5)
224      .ENDM  SET.SACS
225
226      .MACRO  SET.LASTBYTE
227      BIS    @LASTBYTE,CIR(R5)
228      .ENDM  SET.LASTBYTE
229
230      .MACRO  SET.MC
231      BIS    @MC,CIR(R5)
232      .ENDM  SET.MC
233
234      .MACRO  MASTERCLEAR
235      BIS    @BIT5,CIR(R5)
236      BIS    @BIT3,CSR(R4)
237      .ENDM
238      .MACRO  SET.INTENB
239      BIS    @INTENB,CIR(R5)
240      .ENDM  SET.INTENB
241      .MACRO  GO
242      BIS    RBUFEA,CSR(R4)
243      BIS    @1,CSR(R4)
244      .ENDM
245
246      .MACRO  SET.CA
247      BIS    @TCA,SMR(R5)
248      .ENDM  SET.CA
249
250      .MACRO  SET.CS
251      BIS    @TCS,SMR(R5)
252      .ENDM  SET.CS
253
254      .MACRO  SET.GTS
255      BIS    @GTS,SMR(R5)
256      .ENDM  SET.GTS
257
258      .MACRO  SET.SIC
259      BIS    @SIC,SMR(R5)
260      .ENDM  SET.SIC
261
262      .MACRO  CLR.STATECHGE
263      BIC    @STATECHGE,CIR(R5)
264      .ENDM  CLR.STATECHGE
265
266      .MACRO  CLR.DATAACC
267      BIC    @DATAACC,CIR(R5)
268      .ENDM  CLR.DATAACC
269
270      .MACRO  TST.0  ABIT,REG,GOIF0
271      BIT    @ABIT,REG(R5)
272
273      BEQ    GOIF0          ; ABIT CLEAR IN REG ?
274
275      .ENDM  TST.0        ; YES.
276                          ; NO.

```

```

277      .MACRO TEST FE,ER,?L$
278      TST FE                                ;TEST FOR 'FORCE ERROR' SET.
279      BEQ L$                                ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
280      JMP ER                                ;JUMP TO NEXT ERROR.
281      NOP                                  ;IF NO FORCE ERROR THEN JUST CONTINUE.
282      .ENDM
283
284      .MACRO TST.1 ABIT,REG,GOIF1
285      BIT 0ABIT,REG(R5)
286
287      BNE GOIF1                                ; ABIT SET IN REG ?
288
289      .ENDM TST.1                                ; YES.
290
291
292      .MACRO DELAY TIME,?L$
293      MOV 0TIME,DELCNT                        ; DELAY ( TIME US ).
294      DEC DELCNT                                ; BREAK TO MONITOR.
295      BNE L$                                  ; TIMEOUT?
296
297      .ENDM DELAY                                ; NO.
298
299
300      .MACRO WAIT.0 TIME,ABIT,REG,GOIFO,?L$
301      MOV 0TIME,DELCNT                        ; WAIT ( TIME US ) FOR ABIT = 0.
302      BIT 0ABIT,REG(R5)
303
304      BEQ GOIFO                                ; BIT ABIT = 0?
305      BREAK                                  ; YES.
306      DEC DELCNT                                ; NO. BREAK TO MONITOR.
307      BNE L$                                  ; TIMEOUT?
308
309      .ENDM WAIT.0                                ; NO.
310
311
312      .MACRO WAIT.1 TIME,ABIT,REG,GOIF1,?L$
313      MOV 0TIME,DELCNT                        ; WAIT ( TIME US ) FOR ABIT = 1.
314      BIT 0ABIT,REG(R5)
315
316      BNE GOIF1                                ; BIT ABIT = 1?
317      BREAK                                  ; YES.
318      DEC DELCNT                                ; NO. BREAK TO MONITOR.
319      BNE L$                                  ; TIMEOUT?
320
321      .ENDM WAIT.1                                ; NO.
322
323
324      .MACRO WAIT.INT
325      MOV R5,0VR5                            ; SAVE CONTENTS OF R5.
326      EXIT                                    ; WAIT FOR INTERRUPT - EXIT TO MONITOR.
327
328      .ENDM WAIT.INT
329
330      .MACRO ERROR XREF,ECODE
331      JSR PC,ESETUP                            ; SET UP FOR ERROR CALL.
332      MOV 0ECODE,FHRTYP                        ; SET ERROR CODE TYPE.
333      MSGN MSG'XREF'                            ; X-REF ERR CODE & DESCRIPTIVE MSG.
334      HDRR REGTAB                                ; CALL ERROR, PRINT ALL REGS.
335      NOP
336      NOP
    
```

```

334          .ENDM      ERROR
335
336          .MACRO    SCOPE
337          NOP
338          NOP
339          .ENDM      SCOPE
340
341          .MACRO    ERRB      XREF, ECODE
342          JSR      PC, ESETB      ; SET UP FOR ERROR CALL
343          MOV      @ECODE, ERRTYP  ; SET ERROR CODE TYPE
344          MSGN     MSG' XREF'      ; X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
345          HRDR     REGTAB         ; CALL ERROR, PRINT ALL RECS.
346          NOP
347          NOP
348          .ENDM
349
350          .MACRO    COMPARE, BRNCH, ?L$
351          MOV      @400, R3        ; SET UP FOR 256. COMPARES
352          BUF1
353          BUF2
354          L$:      DEC      R3      ; KEEP TRACK
355          BEQ      BRNCH          ; IF DONE GO TO NEXT TEST
356          CMPB    (R2)+, (R1)+    ; COMPARE DATA
357          BEQ      L$             ; IF GOOD DO IT AGAIN
358          .ENDM
359
360          .MACRO    BUF1
361          MOV      RBUFVA, R1
362          .ENDM
363
364          .MACRO    BUF2
365          MOV      RBUFVA, R2
366          ADD     @256., R2
367          .ENDM
    
```

EQUATES

```

370          .SBTTL EQUATES
371
372          ;**
373          ; DEVICE REGISTERS AND BIT DEFINITIONS.
374          ;--
375
376          000000      CIR=    0          ; CONTROL & INTERRUPT REGISTER.
377          000001      CACS=    BIT0         ; SYSTEM CONTROL ACTIVE STATE.
378          000004      LASTBYTE= BIT2         ; LAST BYTE.
379          000040      MC=      BIT5         ; MASTER CLEAR.
380          000100      INTENB=  BIT6         ; INTERRUPT ENABLE.
381          000400      STTECHGE= BIT8         ; STATE CHANGE.
382          010000      ILLMSG=  BIT12        ; ILLEGAL MESSAGE.
383          020000      END=     BIT13        ; END OF BLOCK.
384          100000      DATAACC= BIT15       ; DATA ACCEPTED.
385
386          000002      SMR=    2          ; STATE & MESSAGE REGISTER.
387          000002      TCA=    BIT1         ; TAKE CONTROL ASYNCHRONOUSLY.
388          000001      TCS=    BIT0         ; TAKE CONTROL SYNCHRONOUSLY.
389          000004      GTS=    BIT2         ; GO TO STANDBY.
390          000100      SIC=    BIT6         ; SEND INTERFACE CLEAR.
391          000400      CACS=    BIT8         ; CONTROLLER ACTIVE STATE.
392          001000      CSBS=   BIT9         ; CONTROLLER STANDBY STATE.
393          010000      LACS=   BIT12        ; LISTENER ACTIVE STATE.
394          020000      SIAS=   BIT13        ; INTERFACE CLEAR ACTIVE STATE.
395
396          000004      IOR=    4          ; INPUT & OUTPUT REGISTER.
397
398          000005      IORHB=  5          ; HIGH BYTE OF INPUT/OUTPUT REGISTER.
399          000006      VSR=    6          ; VECTOR SWITCH REGISTER.
400          000000      CSR=    0          ; CONTROL STATUS REGISTER IEC11-B
401          000001      EF=     BIT0         ; ENABLE FUNCTION
402          000010      MCB=    BIT3         ; MASTER CLEAR IEC11-B
403          000100      IE=     BIT6         ; INTERRUPT ENABLE
404
405          000002      BCR=    2          ; BYTE COUNT REGISTER
406
407          000004      BAR=    4          ; BUS ADDRESS REGISTER
408
409          000006      MCR=    6          ; MATCH CHARACTER REGISTER
410
411          ;**
412          ; GENERAL EQUATES.
413          ;--
414
415          177777      ALL=    177777     ; ALL BITS (BITS 15-0).
416          000377      LOBYTE= 377       ; LO ORDER BITS (BITS 7-0).
417          177400      HIBYTE= 177400    ; HI ORDER BITS (BITS 15-8).

```

```

420          ,SBTTL  START - RESTART - TEST
421
425
426 000252   START:
427 000252   RESTRT:
428
429          ;++
430          ; SET UP VECTOR, BR LEVEL, AND R5 AS DEVICE POINTER.
431          ;--
432 000252   GETPA  RBUFVA
      000252 104415 000000' 000124'   GETPA$,BEGIN, RBUFVA           ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
433
434 000260   MOV     VECTOR,R5           ; GET VECTOR ADDRESS.
435 000264   MOV     @INTSVC,(R5)+      ; SET COMPLETION INTERRUPT VECTOR.
436 000270   MOV     BR1,(R5)          ; SET BREAK LEVEL.
437 000274   MOV     ADDR,R5           ; R5 = DEVICE BASE ADDRESS.
438 000300   MOV     R5,R4
439 000302   ADD     @10,R4            ;SET DEFAULT IEC11-B ADDRESS
440 000306   MOV     VECTOR,R2
441 000312   ADD     @4,R2              ;SET DEFAULT IEC11-B VECTOR.
442 000316   MOV     @35,AADR          ;SET DEFAULT IEC11-A IEEE ADDRESS
443 000324   MOV     @36,BADR          ;SET DEFAULT IEC11-B IEEE ADDRESS
444 000332   TST     SR1               ;CHECK FOR OPERATOR SET ADDRESS(IEC11-B)
445 000336   BEQ     1$                ;IF NOT SET CHECK SR2
446 000340   MOV     SR1,R4            ;SET UP OPERATOR SPECIFIED ADDRESS.
447 000344   TST     SR2               ;CHECK FOR OPERATOR SPECIFIED VECTOR(IEC11-B)
448 000350   BEQ     2$                ;IF NOT SET CHECK SR3
449 000352   MOV     SR2,R2            ;SET OPERATOR SPECIFIED VECTOR
450 000356   TST     SR3               ;CHECK FOR OPERATOR SPECIFIED IEEE ADDRESS(IEC11-A).
451 000362   BEQ     3$                ;IF NOT SET CHECK SR4
452 000364   MOV     SR3,AADR          ;SET OPERATOR SPECIFIED IEEE BUS ADDRESS(IEC11-A)
453 000372   TST     SR4               ;CHECK FOR OPERATOR SPECIFIED IEEE ADDRESS(IEC11-B)
454 000376   BEQ     4$                ;IF NOT SET ASSUME DEFAULTS AND CONTINUE.
455 000400   MOV     SR4,BADR          ;SET OPERATOR SPECIFIED IEEE BUS ADDRESS(IEC11-B)
456 000406   NOP                      ;CONTINUE
457 000410   MOV     @INTSVB,(R2)      ;SET INTERRUPT VECTOR
458 000414   MOV     BR2,2(R2)         ;SET B BREAK LEVEL
459 000422   CLR     DATBUF            ; CLEAR THE DATA BUFFER.
460
461          ;++
462          ; TEST DATA TRANSFER WITH INTERRUPT.
463          ;--
464
465 000426   TSTIEA:                   ; SET UP AND TEST THE IEC11-A AND IEC11-B.
466 000426   TEST    FE,ER1              ;TEST FOR 'FORCE ERROR' SET.
      000428 005767 005450             ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
      000432 001402                     ;JUMP TO NEXT ERROR.
      000434 000167 000210             ;IF NO FORCE ERROR THEN JUST CONTINUE.
      000440 000240
467 000442   64$:
468 000442   T1:
      000442 052765 000040 000000     SET,MC                       ; MASTER CLEAR.
      000442 052765 000040 000000     BIS  @MC,CIR(R5)             ; SYSTEM CONTROL ACTIVE STATE.
469 000450   SET,SACS                       ; SEND INTERFACE CLEAR.
      000450 052765 000001 000000     BIS  @SACS,CIR(R5)
470 000456   SET,SIC                       ; (GIVE STATECHGE TIME TO CLEAR).
      000456 052765 000100 000002     BIS  @SIC,SMR(R5)
471 000464   DELAY 100

```

```

000464 012767 000100 005370      MOV    #100,DELCNT      ; DELAY ( 100 US ).
000472 000472      BREAK      ; BREAK TO MONITOR.
000472 104407 000000'      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
000476 104407 000000'      BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
000502 005367 005354      DEC     DELCNT        ; TIMEOUT?
000506 001371      BNE     64$          ; NO.
                                ; YES.
472 000510      CLR.STATECHGE        ; CLEAR INTERRUPT CONDITIONS.
000510 042765 000400 000000      BIC     #STATECHGE,CIR(R5)
473 000516      DELAY    100          ; (GIVE SIAS TIME TO CLEAR).
000516 012767 000100 005336      MOV     #100,DELCNT    ; DELAY ( 100 US ).
000524 000524      BREAK      ; BREAK TO MONITOR.
000524 104407 000000'      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
000530 104407 000000'      BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
000534 005367 005322      DEC     DELCNT        ; TIMEOUT?
000540 001371      BNE     65$          ; NO.
                                ; YES.
474 000542      CLR.STATECHGE        ; CLEAR STATE CHANGE.
000542 042765 000400 000000      BIC     #STATECHGE,CIR(R5)
475 000550      SET.GTS          ; GO TO STANDBY STATE.
000550 052765 000004 000002      BIS     #GTS,SMR(R5)
476 000556      SCOPE
000556 000240      NOP
000560 000240      NOP
477 000562      DELAY    100          ; (GIVE CSBS TIME TO SET).
000562 012767 000100 005272      MOV     #100,DELCNT    ; DELAY ( 100 US ).
000570 000570      BREAK      ; BREAK TO MONITOR.
000570 104407 000000'      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
000574 104407 000000'      BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
000600 005367 005256      DEC     DELCNT        ; TIMEOUT?
000604 001371      BNE     66$          ; NO.
                                ; YES.
478 000606      SET.CS          ; TAKE CONTROL SYNCR.
000606 052765 000001 000002      BIS     #TCS,SMR(R5)
479 000614      WAIT.1  100,ILLMSGE,CIR,T2
000614 012767 000100 005240      MOV     #100,DELCNT    ; WAIT ( 100 US ) FOR ILLMSGE = 1.
000622 032765 010000 000000      BIT     #ILLMSGE,CIR(R5)
                                ; BIT ILLMSGE = 1?
000630 001032      BNE     T2          ; YES.
000632      BREAK      ; NO. BREAK TO MONITOR.
000632 104407 000000'      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
000636 104407 000000'      BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
000642 005367 005214      DEC     DELCNT        ; TIMEOUT?
000646 001365      BNE     67$          ; NO.
                                ; YES.
480 000650      ERROR  46,25        ; NO ILLEGAL MESSAGE.
000650 004767 004026      JSR     PC,ESETUP     ; SET UP FOR ERROR CALL.
000654 012767 000025 177224      MOV     #25,ERRTYP    ; SET ERROR CODE TYPE.
000662      MSGN    MSG46        ; X-REF ERR CODE & DESCRIPTIVE MSG.
000662 104403 000000' 006134'      MSGN$,BEGIN,MSG46    ;ASCII MESSAGE CALL WITH COMMON HEADER
000670      HRDR    REGTAB      ; CALL ERROR, PRINT ALL REGS.
                                ;*****
000670 104405 000000' 006110'      HRDR$,BEGIN,REGTAB
                                ;*****
000676 000240      NOP
000700 000240      NOP
481 000702      TEST   FE,ER2

```

```

000702 005767 005174          TST      FE          ;TEST FOR 'FORCE ERROR' SET.
000706 001402                BEQ      64$         ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
000710 000167 000114          JMP      ER2        ;JUMP TO NEXT ERROR.
000714 000240                NOP                    ;IF NO FORCE ERROR THEN JUST CONTINUE.
482 000716                T2:
483 000716 042765 177400 000000 BIC      #HIBYTE,CIR(R5) ; CLEAR INTERRUPT CONDITIONS.
484 000724                SET.CA          ; TAKE CONTROL ASYNCR.
000724 052765 000002 000002 BIS      #TCA,SMR(R5)
485 000732                DELAY 100          ; (GIVE STATECHGE & CACS TIME TO SET).
000732 012767 000100 005122 MOV      #100,DELCNT ; DELAY ( 100 US ).
000740                64$:
000740 104407 000000'        BREAK ; BREAK TO MONITOR.
000744 104407 000000'        BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
000750 005367 005106        DEC      DELCNT      ; THEN CONTINUE AT NEXT INSTRUCTION.
000754 001371                BNE      64$         ; TIMEOUT?
                                ; NO.
                                ; YES.
486 000756                SCOPE
000756 000240                NOP
000760 000240                NOP
487 000762 042765 177400 000000 BIC      #HTBYTE,CIR(R5) ; CLEAR INTERRUPT BITS.
488 000770 012767 001076' 005072 MOV      #15,IRTN     ; SET UP INTERRUPT RETURN ADDRESS.
489 000776                SET.INTENB      ; ENABLE INTERRUPT.
000776 052765 000100 000000 BIS      #INTENB,CIR(R5)
490 001004 016700 005070        MOV      LRAS,RO     ;SET UP LISTENER BASE ADDRESS
491 001010 066700 005072        ADD      AADR,RO     ;ADD IN IEEE BUS ADDRESS(IEC11-A)
492 001014 110065 000004        MOV.B   RO,IOR(R5)  ; SEND LISTENER ADDRESS.
493 001020                WAIT.INT
001020 010567 177050        MOV      R5,SVRS    ; SAVE CONTENTS OF R5.
001024                EXIT ; WAIT FOR INTERRUPT - EXIT TO MONITOR.
001024 104400 000000'        EXIT$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
494 001030                ER2:
001030 004767 003646          JSR      PC,ESETUP   ; NO INTERRUPT TO LISTENER ADDRESSING.
001034 012767 000023 177044 MOV      #23,ERRTYP ; SET UP FOR ERROR CALL.
001042                MSGN MSG50          ; SET ERROR CODE TYPE.
001042 104403 000000' 006140' MSGN$,BEGIN,MSG50 ; X-REF ERR CODE & DESCRIPTIVE MSG.
001050                HRDR REGTAB ; ASCII MESSAGE CALL WITH COMMON HEADER
                                ; CALL ERROR, PRINT ALL REGS.
                                ;*****
001050 104405 000000' 006110' HRDR$,BEGIN,REGTAB ;
                                ;*****
001056 000210                NOP
001060 000240                NOP
495 001062                TEST      .E,ER3
001062 005767 005014          TST      FE          ;TEST FOR 'FORCE ERROR' SET.
001066 001402                BEQ      64$         ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
001070 000167 000054          JMP      ER3        ;JUMP TO NEXT ERROR.
001074 000240                64$:
496 001076                T3:
497 001076                T4:
498 001076                SCOPE
001076 000240                NOP
001100 000240                NOP
499 001102 042765 177400 000000 BIC      #HIBYTE,CIR(R5) ; CLEAR INTERRUPT BITS.
500 001110 012767 001216' 004752 MOV      #T5,IRTN     ; SET UP INTERRUPT RETURN ADDRESS.
501 001116                SET.INTENB      ; ENABLE INTERRUPT.
001116 052765 000100 000000 BIS      #INTENB,CIR(R5)
502 001124 016700 004746        MOV      TBAS,RO     ;SET UP TALKER BASE ADDRESS

```

```

503 001130 066700 004752      ADD      AADR,R0          ;ADD IN IEEE BUS ADDRESS(IEC11-A)
504 001134 110065 000004      MOVB     RO,IOR(R5)      ; ADDRESS RO AGAIN.
505 001140      WAIT,INT
   001140 010567 176730      MOV      R5,SVR5        ; SAVE CONTENTS OF R5.
   001144      EXIT        ; WAIT FOR INTERRUPT - EXIT TO MONITOR.
   001144 104400 000000'      EXIT$,BEGIN           ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

506 001150      ER3:      ERROR  53,23          ; NO INTERRUPT TO TALKER ADDRESSING.
   001150 004767 003526      JSR      PC,ESETUP      ; SET UP FOR ERROR CALL.
   001154 012767 000023 176724  MOV      #23,ERRTYP     ; SET ERROR CODE TYPE.
   001162      MSGN      MSG53          ; X-REF ERR CODE & DESCRIPTIVE MSG.
   001162 104403 000000' 006150'  MSGN$,BEGIN,MSG53      ;ASCII MESSAGE CALL WITH COMMON HEADER
   001170      HRDR      REGTAB        ; CALL ERROR, PRINT ALL REGS.
   ;*****
   001170 104405 000000' 006110'  HRDR$,BEGIN,REGTAB    ;
   ;*****
   001176 000240      NOP
   001200 000240      NOP
507 001202      TEST      FE,ERA          ;TEST FOR 'FORCE ERROR' SET.
   001202 005767 004674      TST      FE              ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
   001206 001402      BEQ      64$           ;JUMP TO NEXT ERROR.
   001210 000167 000044      JMP      ERA             ;IF NO FORCE ERROR THEN JUST CONTINUE.
   001214 000240      64$:      NOP
508 001216      T5:
509 001216      SET,GTS          ; GO TO STANDBY STATE.
   001216 052765 000004 000002  BIS      #GTS,SMR(R5)
510 001224      DELAY 100          ; (GIVE CSBS TIME TO SET).
   001224 012767 000100 004630  MOV      #100,DELCNT    ; DELAY ( 100 US ).
   001232      64$:      BREAK          ; BREAK TO MONITOR.
   001232 104407 000000'      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
   001236 104407 000000'      BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
   001242 005367 004614      DEC      DELCNT        ; TIMEOUT?
   001246 001371      BNE      64$           ; NO.
   ; YES.

511 001250      TST,1      LACS,SMR,T6
   001250 032765 010000 000002  BIT      @LACS,SMR(R5)
   ; LACS SET IN SMR ?
   ; YES.
   ; NO.
   001256 001023      BNE      T6
512 001260      ER4:      ERROR  52,25          ; NO LACS.
   001260 004767 003416      JSR      PC,ESETUP      ; SET UP FOR ERROR CALL.
   001264 012767 000025 176614  MOV      #25,ERRTYP     ; SET ERROR CODE TYPE.
   001272      MSGN      MSG52          ; X-REF ERR CODE & DESCRIPTIVE MSG.
   001272 104403 000000' 006144'  MSGN$,BEGIN,MSG52      ;ASCII MESSAGE CALL WITH COMMON HEADER
   001300      HRDR      REGTAB        ; CALL ERROR, PRINT ALL REGS.
   ;*****
   001300 104405 000000' 006110'  HRDR$,BEGIN,REGTAB    ;
   ;*****
   001306 000240      NOP
   001310 000240      NOP
513 001312      TEST      FE,ER5          ;TEST FOR 'FORCE ERROR' SET.
   001312 005767 004564      TST      FE              ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
   001316 001402      BEQ      64$           ;JUMP TO NEXT ERROR.
   001320 000167 000110      JMP      ER5            ;IF NO FORCE ERROR THEN JUST CONTINUE.
   001324 000240      64$:      NOP
514 001326      T6:
515 001326 112765 177777 000004  MOVB     #ALL,IOR(R5)    ; SEND A DATA BYTE.

```

START RESTART TEST

```

516 001334          DELAY 100          ; ( GIVE DATAACC TIME TO SET ).
    001334 012767 000100 004520          MOV   #100,DELCNT          ; DELAY ( 100 US ).
    001342          BREAK          ; BREAK TO MONITOR.
    001342 104407 000000' 64$:          BREAK$,BEGIN          ; TEMPORARY RETURN TO MONITOR....
    001346 104407 000000'          BREAK$,BEGIN          ; THEN CONTINUE AT NEXT INSTRUCTION.
    001352 005367 004504          DEC   DELCNT          ; TIMEOUT?
    001356 001371          BNE   64$          ; NO.
                                          ; YES.
517 001360          CLR,DATAACC          ; CLEAR DATA ACCEPTED.
    001360 042765 100000 000000          BIC   #DATAACC,CIR(R5)
518 001366          CLR,STATECHGE          ; CLEAR STATE CHANGE.
    001366 042765 000400 000000          BIC   #STATECHGE,CIR(R5)
519 001374 105365 000004          DECB  IOR(R5)          ; SEND NEXT DATA BYTE.
520 001400          DELAY 100
    001400 012767 000100 004454          MOV   #100,DELCNT          ; DELAY ( 100 US ).
    001406          BREAK          ; BREAK TO MONITOR.
    001406 104407 000000' 65$:          BREAK$,BEGIN          ; TEMPORARY RETURN TO MONITOR....
    001412 104407 000000'          BREAK$,BEGIN          ; THEN CONTINUE AT NEXT INSTRUCTION.
    001416 005367 004440          DEC   DELCNT          ; TIMEOUT?
    001422 001371          BNE   65$          ; NO.
                                          ; YES.
521 001424          TST,0 DATAACC,CIR,T7
    001424 032765 100000 000000          BIT   #DATAACC,CIR(R5)
                                          ; DATAACC CLEAR IN CIR ?
    001432 001423          BEQ   T7          ; YES.
                                          ; NO.
522 001434          ERROR 54,44          ; DATA ACCEPTED TO SECOND DATA BYTE.
    001434 004757 003242          JSR   PC,ESETUP          ; SET UP FOR ERROR CALL.
    001440 012767 000044 176440          MOV   #44,ERRTYP          ; SET ERROR CODE TYPE.
    001446          MSGN MSG54          ; X-REF ERR CODE & DESCRIPTIVE MSG.
    001446 104403 000000' 006154'          MSGN$,BEGIN,MSG54          ; ASCII MESSAGE CALL WITH COMMON HEADER
    001454          HRDR REGTAB          ; CALL ERROR, PRINT ALL REGS.
    001454 104405 000000' 006110'          HRDR$,BEGIN,REGTAB          ; *****
    001462          NOP
    001464          NOP
523 001466          TEST FE,ER6
    001466 005767 004410          TST   FE          ; TEST FOR 'FORCE ERROR' SET.
    001472 001402          BEQ   64$          ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
    001474 000167 000030          JMP   ER6          ; JUMP TO NEXT ERROR.
    001500          NOP          ; IF NO FORCE ERROR THEN JUST CONTINUE.
524 001502          T7:
525 001502 016567 000004 004364          MOV   IOR(R5),GOT          ; GET IOR CONTENT.
526 001510 000367 004360          SWAB  GOT          ;
527 001514 126727 004354 000377          CMPB  GOT,#377          ; DATA OK?
528 001522 001425          BEQ   T8          ; YES.
529 001524 000367 004344          SWAB  GOT          ; NO, REBUILD DATA PATTERN READ.
530 001530          ERROR 55,1          ; DATA CHECK ON IEC TALKER OR LISTENER.
    001530 004767 003146          JSR   PC,ESETUP          ; SET UP FOR ERROR CALL.
    001534 012767 000001 176344          MOV   #1,ERRTYP          ; SET ERROR CODE TYPE.
    001542          MSGN MSG55          ; X-REF ERR CODE & DESCRIPTIVE MSG.
    001542 104403 000000' 006160'          MSGN$,BEGIN,MSG55          ; ASCII MESSAGE CALL WITH COMMON HEADER
    001550          HRDR REGTAB          ; CALL ERROR, PRINT ALL REGS.
    001550 104405 000000' 006110'          HRDR$,BEGIN,REGTAB          ; *****

```

```

001556 000240      NOP
001560 000240      NOP
531 001562      TEST    FE,ER7
001562 005767 004314  TST    FE                ; TEST FOR 'FORCE ERROR' SET.
001566 001402      BEQ    64$                ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
001570 000167 000012  JMP    ER7                ; JUMP TO NEXT ERROR.
001574 000240      NOP                ; IF NO FORCE ERROR THEN JUST CONTINUE.
532 001576      64$:
533 001576      T8:
001576 032765 100000 000000  TST.1  DATAACC,CIR,T9
BIT    @DATAACC,CIR(R5)
                                ; DATAACC SET IN CIR ?
                                ; YES.
                                ; NO.
534 001606      ER7:  ERROR   56,25
001606 004767 003070  JSR    PC,ESETUP          ; NO SECOND "DATA ACCEPTED".
001612 012767 000025 176266  MOV    @25,ERRTYP        ; SET UP FOR ERROR CALL.
001620      MSGN   MSG56          ; SET ERROR CODE TYPE.
001620 104403 000000' 006164'  MSGN$,BEGIN,MSG56        ; X-REF ERR CODE & DESCRIPTIVE MSG.
001626      HRDR   REGTAB          ; ASCII MESSAGE CALL WITH COMMON HEADER
                                ; CALL ERROR, PRINT ALL REGS.
001626 104405 000000' 006110'  HRDR$,BEGIN,REGTAB
                                ; *****
                                ; *****
001634 000240      NOP
001636 000240      NOP
535 001640      TEST    FE,ER8
001640 005767 004236  TST    FE                ; TEST FOR 'FORCE ERROR' SET.
001644 001402      BEQ    64$                ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
001646 000167 000030  JMP    ER8                ; JUMP TO NEXT ERROR.
001652 000240      NOP                ; IF NO FORCE ERROR THEN JUST CONTINUE.
536 001654      64$:
537 001654      T9:
001654 016567 000004 004212  MOV    IOR(R5),GOT        ; GET SECOND DATA BYTE.
538 001662      SWAB   GOT
001662 000367 004206  CMPB   GOT,@376          ; IS SECOND DATA BYTE OK?
539 001666      CMPB   GOT,@376
001674 001425      BEQ    T10                ; YES.
540 001674      SWAB   GOT
001676 000367 004172  SWAB   GOT                ; NO, REBUILD PATTERN READ.
541 001676      ERROR  57,1          ; DATA CHECK ON SECOND BYTE.
542 001702      ER8:  JSR    PC,ESETUP          ; SET UP FOR ERROR CALL.
001702 004767 002774  JSR    PC,ESETUP          ; SET ERROR CODE TYPE.
001706 012767 000001 176172  MOV    @1,ERRTYP        ; X-REF ERR CODE & DESCRIPTIVE MSG.
001714      MSGN   MSG57          ; ASCII MESSAGE CALL WITH COMMON HEADER
001714 104403 000000' 006170'  MSGN$,BEGIN,MSG57        ; CALL ERROR, PRINT ALL REGS.
001722      HRDR   REGTAB
                                ; *****
                                ; *****
001722 104405 000000' 006110'  HRDR$,BEGIN,REGTAB
                                ; *****
                                ; *****
001730 000240      NOP
001732 000240      NOP
543 001734      TEST    FE,ER9
544 001734      T8:  TST    FE                ; TEST FOR 'FORCE ERROR' SET.
001734 005767 004142  TST    FE                ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
001740 001402      BEQ    64$                ; JUMP TO NEXT ERROR.
001742 000167 000072  JMP    ER9                ; IF NO FORCE ERROR THEN JUST CONTINUE.
001746 000240      64$:  NOP
545      ;**
546      ; RANDOM TALKER AND LISTENER DATA TEST.
547      ;
548      ; THIS TEST CAN NOT BE CALLED SEPARATELY, DUE TO THE NECESSARY,
549      ; SETUP TO BRING THE BUS INTO THIS STATE.

```

START - RESTART - TEST

```

550                                     ;
551                                     ;--
552
553 001750                               T10:
554 001750 005067 004110                 CLR     RANMOD                ; DISABLE RANDOM MODE.
555 001754 005067 004112                 CLR     DATBUF                ; CLEAR DATA BUFFER.
556 001760 042765 177400 000000         BIC     @HIBYTE,CIR(R5)      ; CLEAR INTERRUPT BITS.
557 001766                               SCOPE
    001766 000240                       NOP
    001770 000240                       NOP
558 001772 012767 002106' 004070         MOV     @T12,IRTN            ; SET INTERRUPT RETURN ADDRESS.
559 002000 012767 001000 004060         MOV     @1000,LOOP          ; INIT LOOP COUNTER.
560 002006                               T11:
561 002006 042765 177400 000000         BIC     @HIBYTE,CIR(R5)      ; CLEAR INTERRUPT BITS.
562 002014                               SET.INTENB                    ; ENABLE INTERRUPTS.
    002014 052765 000100 000000         BIS     @INTENB,CIR(R5)
563 002022 016765 004044 000004         MOV     DATBUF,IOR(R5)      ; SEND A DATA BYTE.
564 002030                               WAIT.INT
    002030 010567 176040                 MOV     R5,SVR5              ; SAVE CONTENTS OF R5.
    002034                               EXIT                            ; WAIT FOR INTERRUPT - EXIT TO MONITOR.
    002034 104400 000000'                 EXIT$,BEGIN                  ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

565 002040                               ER9:
    002040 004767 002636                 JSR     PC,ESETUP            ; NO INTERRUPT TO DATA SENDING.
    002044 012767 000023 176034         MOV     @23,ERRTYP          ; SET UP FOR ERROR CALL.
    002052                               MSGN MSG60                    ; SET ERROR CODE TYPE.
    002052 104403 000000' 006174'       MSGN$,BEGIN,MSG60           ; X-REF ERR CODE & DESCRIPTIVE MSG.
    002060                               HRDR REGTAB                  ; ASCII MESSAGE CALL WITH COMMON HEADER
    002060 104405 000000' 006110'       HRDR$,BEGIN,REGTAB         ; CALL ERROR, PRINT ALL REGS.
    002066 000240                       ;*****
    002070 000240                       ;*****
    002070                               NOP
    002070                               NOP

566
567 002072                               T12:
    002072 005767 004004                 TEST    FE,ER10              ; TEST FOR 'FORCE ERROR' SET.
    002076 001402                         TST     FE
    002100 000167 000030                 BEQ     64$                  ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
    002104 000240                         JMP     ER10                 ; JUMP TO NEXT ERROR.
    002104 000240                         NOP                          ; IF NO FORCE ERROR THEN JUST CONTINUE.

568 002106                               64$:
569 002106                               T12:
570 002106 016567 000004 003760         MOV     IOR(R5),GOT          ; READ BACK DATA BYTE.
571 002114 105067 003754                 CLRB   GOT                  ; PREPARE DATA.
572 002120 000367 003750                 SWAB   GOT
573 002124 026767 003744 003740         CMP     GOT,DATBUF           ; DATA OK?
574 002132 001423                         BEQ     T14                  ; YES.
575 002134                               ER10:
    002134 004767 002542                 ERROR  62,1                  ; DATA CHECK ON IEC BUS.
    002140 012767 000001 175740         JSR     PC,ESETUP            ; SET UP FOR ERROR CALL.
    002146                               MOV     @1,ERRTYP           ; SET ERROR CODE TYPE.
    002146 104403 000000' 006200'       MSGN   MSG62                ; X-REF ERR CODE & DESCRIPTIVE MSG.
    002154                               MSGN$,BEGIN,MSG62           ; ASCII MESSAGE CALL WITH COMMON HEADER
    002154 104405 000000' 006110'       HRDR   REGTAB                ; CALL ERROR, PRINT ALL REGS.
    002162 000240                       ;*****
    002164 000240                       ;*****
    002164                               NOP
    002164                               NOP

```

```

576
577 002166          TEST    FE,ER11
      002166 005767 003710    TST     FE
      002172 001402          BEQ     64$
      002174 000167 000142    JMP     ER11
      002200 000240          NOP
      64$:
      T14:
578 002202          TST     RANMOD
579 002202 005767 003656    BNE     T15
580 002206 001011          INC     DATBUF
581 002210 005267 003656    CMP     DATBUF,0400
582 002214 026727 003652 000400 BNE     T11
583 002222 001271          MOV     0-1,RANMOD
584 002224 012767 177777 003632
585 002232          RAND
586 002232          RAND$,BEGIN
      002232 104417 000000'    MOV     RANNUM,DATBUF
587 002236 016767 175612 003626    BIC     0177400,DATBUF
588 002244 042767 177400 003620    DEC     LOOP
589 002252 005367 003610    BNE     T11
590 002256 001253          CLR     RANMOD
591 002260 005067 003600    MOV     022,DATBUF
592 002264 012767 000222 003600    SET.LASTBYTE
593 002272          BIS     0LASTBYTE,CIR(R5)
      002272 052765 000004 000000    MOV     DATBUF,IUR(R5)
594 002300 016765 003566 000004    WAIT.1 100,DATAACC,CIR,T16
595 002306          MOV     0100,DELCNT
      002306 012767 000100 003546    BIT     0DATAACC,CIR(R5)
      002314 032765 100000 000000 64$:
      BNE     T16
      BREAK
      BREAK$,BEGIN
      BREAK$,BEGIN
      DEC     DELCNT
      BNE     64$
      ER11:
596 002342          ERROR   63,25
      002342 004767 002334    JSR     PC,ESETUP
      002346 012767 000025 175532    MOV     025,ERRTYP
      002354          MSGN    MSG63
      002354 104403 000000' 006204'  MSGN$,BEGIN,MSG63
      002362          HRDR    REGTAB
      002362 104405 000000' 006110'  HRDR$,BEGIN,REGTAB
      002370 000240          NOP
      002372 000240          NOP
597
598 002374          TEST    FE,FR12
      002374 005767 003502    TST     FE
      002400 001402          BEQ     64$
      002402 000167 000012    JMP     ER12
      002406 000240          NOP
      64$:
      T16:
599 002410          TST.1  END,CIR,T17
600 002410 032765 020000 000000    BIT     0END,CIR(R5)

```

```

; TEST FOR 'FORCE ERROR' SET.
; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
; JUMP TO NEXT ERROR.
; IF NO FORCE ERROR THEN JUST CONTINUE.

```

```

; RANDOM MODE?
; YES.
; NO. BUILD NEXT DATA PATTERN.
; ALL DATA (0-377) TRIED?
; NO.
; YES. SWITCH TO RANDOM MODE.

```

```

; GET A RANDOM NUMBER (PATTERN).
; MOVE PATTERN ONTO STACK.
; BUILD A BYTE.
; BUMP LOOP COUNT. DONE?
; NO.
; SWITCH BACK TO NORMAL MODE.
; PREPARE LAST DATA PATTERN.
; INDICATE "LAST BYTE".

```

```

; LOAD IT.
; WAIT ( 100 US ) FOR DATAACC = 1.

```

```

; BIT DATAACC = 1?
; YES.
; NO. BREAK TO MONITOR.
; TEMPORARY RETURN TO MONITOR....
; THEN CONTINUE AT NEXT INSTRUCTION.
; TIMEOUT?
; NO.
; YES.

```

```

; NO DATA ACC WHEN SENDING LAST BYTE.
; SET UP FOR ERROR CALL.
; SET ERROR CODE TYPE.
; X-REF ERR CODE & DESCRIPTIVE MSG.
; ASCII MESSAGE CALL WITH COMMON HEADER
; CALL ERROR, PRINT ALL REGS.

```

```

;*****
;*****
;*****

```

```

; TEST FOR 'FORCE ERROR' SET.
; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
; JUMP TO NEXT ERROR.
; IF NO FORCE ERROR THEN JUST CONTINUE.

```

```

; END SET IN CIR ?

```

```

002416 001023          BNE      T17              ; YES.
601 002420          ER12:  ERROR    64,25          ; NO.
002420 004767 002256  JSR      PC,ESETUP        ; NO "END" TO "LAST BYTE".
002424 012767 000025 175454  MOV      #25,ERRTYP      ; SET UP FOR ERROR CALL.
002432          MSGN    MSG64          ; SET ERROR CODE TYPE.
002432 104403 000000' 006210' MSGN$,BEGIN,MSG64      ; X-REF ERR CODE & DESCRIPTIVE MSG.
002440          HRDR    REGTAB          ; ASCII MESSAGE CALL WITH COMMON HEADER
; CALL ERROR, PRINT ALL REGS.
;*****
002440 104405 000000' 006110' HRDR$,BEGIN,REGTAB ;
;*****
002446 000240          NOP
002450 000240          NOP
602 002452          TEST     FE,ER13
603 002452 005767 003424  TST     FE              ; TEST FOR 'FORCE ERROR' SET.
002456 001402          BEQ     64$          ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
002460 000167 000030  JMP     ER13           ; JUMP TO NEXT ERROR.
002464 000240          NOP          ; IF NO FORCE ERROR THEN JUST CONTINUE.
604 002466          64$:
T17:          MOV     IOR(R5),GOT        ; READ BACK DATA BYTE.
605 002466 016567 000004 003400  CLRB   GOT             ;
606 002474 105067 003374          SWAB   GOT             ; PREPARE FOR CHECK.
607 002500 000367 003370          CMP    GOT,#222        ; DATA OK?
608 002504 025727 003364 000222  BEQ    TST4B          ; YES.
609 002512 001123          ER13:  ERROR    65,1          ; DATA CHECK ON "LAST BYTE".
610 002514          JSR     PC,ESETUP        ; SET UP FOR ERROR CALL.
002514 004767 002162 175360  MOV     #1,ERRTYP      ; SET ERROR CODE TYPE.
002520 012767 000001 006214' MSGN    MSG65          ; X-REF ERR CODE & DESCRIPTIVE MSG.
002526 104403 000000' 006214' MSGN$,BEGIN,MSG65      ; ASCII MESSAGE CALL WITH COMMON HEADER
002534          HRDR    REGTAB          ; CALL ERROR, PRINT ALL REGS.
;*****
002534 104405 000000' 006110' HRDR$,BEGIN,REGTAB ;
;*****
002542 000240          NOP
002544 000240          NOP
611 002546          TEST     FE,ER14
002546 005767 003330  TST     FE              ; TEST FOR 'FORCE ERROR' SET.
002552 001402          BEQ     64$          ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
002554 000167 000154  JMP     ER14           ; JUMP TO NEXT ERROR.
002560 000240          NOP          ; IF NO FORCE ERROR THEN JUST CONTINUE.
612          ;TEST IEC11-B --- IEC11-B IS TALKER AND IEC11-A IS LISTENER
613          ;
614          ;
615          ;
616 002562          TST4B:  MASTERCLEAR        ;CLEAR IEC WORLD
002562 052765 000040 000000  BIS    #BIT5,CIR(R5)
002570 052764 000010 000000  BIS    #BIT3,CSR(R4)
617 002576 005003          CLR    R3              ;SET UP COUNTER
618 002600          BUF1          ;GET BUFFER ADDRESS
002600 016701 175320  MOV    RBUFVA,R1
619 002604 110321          1$:  MOVB   R3,(R1)        ;LOAD CHARS IN BUFFER
620 002606 005203          INC    R3
621 002610 022703 000400  CMP    #400,R3        ;CHECK FOR END OF BUFFER
622 002614 001373          BNE    1$
623
624 002616          SET,SACS          ;SET SACS

```

```

625 002616 052765 000001 000000      BIS      #SACS,CIR(R5)
      002624 052765 000100 000002      SET,SIC      ;SET SIC
626 002632 012767 000050 003222      DELAY     50      ;GIVE IT SOME TIME
      002640 012767 000050 003222      MOV       #50,DELCNT ; DELAY ( 50 US ).
      002640 104407 000000' 64$:   BREAK      ; BREAK TO MONITOR.
      002644 104407 000000'      BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
      002650 005367 003206      BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
      002654 001371      DEC       DELCNT ; TIMEOUT?
      ; NO.
      ; YES.
627 002656 042765 177400 000000 TB2:   BIC       #HIBYTE,CIR(R5) ;CLEAR INTERRUPTABLE BITS
628 002664 016700 003206      MOV       TBAS,RO ;SET TALKER BASE ADDRESS
629 002670 066700 003210      ADD      BADR,RO ;ADD IN IEC11B ADDRESS
630
631
632 002674 110065 000004      MOVB     RO,IOR(R5) ;SET INDIVIDUAL TALKER
633 002700 012767 000050 003154      WAIT.1  50,DATAACC,CIR,C1 ;SEND IT DOWN IEC BUS
      002706 032765 100000 000000 64$:   MOV       #50,DELCNT ; HAS INFO BEEN ACCEPTED?
      ; WAIT ( 50 US ) FOR DATAACC = 1.
      ; BIT DATAACC = 1?
      ; YES.
      ; NO. BREAK TO MONITOR.
      002714 001032      BNE      C1 ;TEMPORARY RETURN TO MONITOR....
      002716 104407 000000'      BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
      002722 104407 000000'      BREAK$,BEGIN ; TIMEOUT?
      002726 005367 003130      DEC       DELCNT ; NO.
      002732 001365      BNE      64$ ; YES.
634 002734 004767 002012 175140 ER14:  ERRB     66,25 ;ERROR-NO DATA ACCEPTED
      002740 012767 000025 175140      JSR      PC,ESETB ; SET UP FOR ERROR CALL
      002746 104403 000000' 006220'  MOV       #25,ERRTYP ; SET ERROR CODE TYPE
      002754 104405 000000' 006110'  MSGN     MSG66 ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
      ;ASCII MESSAGE CALL WITH COMMON HEADER
      ; CALL ERROR, PRINT ALL REGS.
      ;*****
      HRDR$,BEGIN,REGTAB ;
      ;*****
      002762 000240      NOP
      002764 000240      NOP
635 002766 005767 003110      TEST     FE,ER15 ;TEST FOR 'FORCE ERROR' SET.
      002772 001402      TST      FE ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
      002774 000167 000060      BEQ     64$ ;JUMP TO NEXT ERROR.
      003000 000240      JMP     ER15 ; IF NO FORCE ERROR THEN JUST CONTINUE.
636 003002 042765 100000 000000 C1:   CLR,DATAACC ;CLEAR DATA ACCEPT BIT
      003002 016700 003064      BIC     #DATAACC,CIR(R5)
637 003010 066700 003066      MOV     LBAS,RO ;SET LISTENER BASE ADDRESS
638 003014 110065 000004      ADD     AADR,RO ;ADD IN IEC11A ADDRESS
639
640 003020 012767 000050 003030      MOVB     RO,IOR(R5) ;SET INDIVIDUAL LISTENER
641 003024 032765 100000 000000 64$:   WAIT.1  50,DATAACC,CIR,C2 ;SEND ADDR DOWN IEC BUS
      003032 001032      MOV     #50,DELCNT ;DATA ACCEPTED?
      ; WAIT ( 50 US ) FOR DATAACC = 1.
      ; BIT DATAACC = 1?
      ; YES.
      ; NO. BREAK TO MONITOR.
      003040 001032      BNE     C2 ;TEMPORARY RETURN TO MONITOR....
      003042 104407 000000'      BREAK$,BEGIN

```

```

003046 104407 000000'          BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
003052 005367 003004          DEC      DELCNT          ; TIMEOUT?
003056 001365          BNE      64$          ; NO.
                                ; YES.
642 003060          ER15:  ERRB      66,25          ;ERROR-NO DATA ACCEPTED
003060 004767 001666          JSR      PC,ESETB      ; SET UP FOR ERROR CALL
003064 012767 000025 175014    MOV      #25,ERRTYP    ; SET ERROR CODE TYPE
003072          MSGN      MSG66          ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
003072 104403 000000' 006220'  MSGN$,BEGIN,MSG66    ;ASCII MESSAGE CALL WITH COMMON HEADER
003100          HRDR      REGTAB          ; CALL ERROR, PRINT ALL REGS.
                                ;*****
003100 104405 000000' 006110'  HRDR$,BEGIN,REGTAB  ;
                                ;*****
003106 000240          NOP
003110 000240          NOP
643 003112          TEST     FE,ER16
003112 005767 002764          TST     FE              ;TEST FOR 'FORCE ERROR' SET.
003116 001402          BEQ     64$          ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
003120 000167 000146          JMP     ER15          ;JUMP TO NEXT ERROR.
003124 000240          64$:  NOP          ;IF NO FORCE ERROR THEN JUST CONTINUE.
644 003126          C2:   CLR.DATAACC          ;YES THEN CLEAR IT.
003126 042765 100000 000000    BIC     #DATAACC,CIR(R5)
645 003134          SET.GTS          ;GO TO STANDBY.
003134 052765 000004 000002    BIS     #GTS,SMR(R5)
646 003142          SCOPE
003142 000240          NOP
003144 000240          NOP
647 003146 042764 177400 000000  BIC     #HIBYTE,CSR(R4) ;CLEAR INTERRUPTABLE BITS
648 003154          BUF1
003154 016701 174744          MOV     RBUFVA,R1
649 003160 016764 174742 000004    MOV     RBUFPA,BAR(R4) ;SET BUFFER ADDRESS
650 003166 012700 000377          MOV     #377,R0
651 003172 005400          NEG     R0
652 003174 010064 000002          MOV     R0,BCR(R4)    ;SET BYTE COUNT REG
653 003200          BUF2
003200 016702 174720          MOV     RBUFVA,R2
003204 062702 000400          ADD     #256.,R2
654 003210 012767 003542' 002642    MOV     #LB,MYRTN    ;SET INTERRUPT RETURN
655 003216          GO          ;START SUCKING DATA
003216 056764 174706 000000    BIS     RBUFEA,CSR(R4)
003224 052764 000001 000000    BIS     #1,CSR(R4)
656 003232 005003          CLR     R3          ;INIT COUNTER
657 003234 000240          TDAT:  NOP          ;GIVE IT SOME TIME
658 003236          WAIT.1 50,DATAACC,CIR,RSET ;DATA ACCEPTED?
003236 012767 000050 002616    MOV     #50,DELCNT    ; WAIT ( 50 US ) FOR DATAACC = 1.
003244 032765 100000 000000    64$:  BIT     #DATAACC,CIR(R5)
                                ; BIT DATAACC = 1?
                                ; YES.
003252 001032          BNE     RSET
003254          BREAK
003254 104407 000000'          BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR...
003260 104407 000000'          BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
003264 005367 002572          DEC     DELCNT
003270 001365          BNE     64$          ; TIMEOUT?
                                ; NO.
                                ; YES.
659 003272          ER16:  ERRB      66,25          ;NO DATA ACCEPTED
003272 004767 001454          JSR     PC,ESETB      ; SET UP FOR ERROR CALL
003276 012767 000025 174602    MOV     #25,ERRTYP    ; SET ERROR CODE TYPE

```

```

003304 MSGN MSG66 ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
003304 104403 000000' 006220' MSGN$,BEGIN,MSG66 ;ASCII MESSAGE CALL WITH COMMON HEADER
003312 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
003312 104405 000000' 006110' HRDR$,BEGIN,REGTAB ;
;*****
NOP
NOP
660 003324 TEST FE,ER17
003324 005767 002552 TST FE ;TEST FOR 'FORCE ERROR' SET.
003330 001402 BEQ 64$ ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
003332 000167 000024 JMP ER17 ;JUMP TO NEXT ERROR.
003336 000240 64$: NOP ;IF NO FORCE ERROR THEN JUST CONTINUE.
661 003340 RSET: CLR.DATAACC ;GET READY FOR NEXT XFER
003340 042765 100000 000000 BIC @DATAACC,CIR(R5)
662 003346 116522 000005 MOV# IOR#B(R5),(R2)+ ;SAVE DATA FROM LISTENER
663 003352 005203 INC R3 ;ADD XFER TO COUNTER
664 003354 022703 000400 CMP #400,R3 ;RIGHT NO. OF XFERS?
665 003360 001023 BNE BT ;SHOULD NEVER GET HERE!!
666 003362 ER17: ERRB 71,23 ;BYTE COUNTER NEVER OVRFLD
003362 004767 001364 JSR PC,ESETB ; SET UP FOR ERROR CALL
003366 012767 000023 174512 MOV #23,ERRTYP ; SET ERROR CODE TYPE
003374 MSGN MSG71 ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
003374 104403 000000' 006244' MSGN$,BEGIN,MSG71 ;ASCII MESSAGE CALL WITH COMMON HEADER
003402 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
003402 104405 000000' 006110' HRDR$,BEGIN,REGTAB ;
;*****
NOP
NOP
667 003414 TEST FE,ER18
003414 005767 002462 TST FE ;TEST FOR 'FORCE ERROR' SET.
003420 001402 BEQ 64$ ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
003422 000167 000046 JMP ER18 ;JUMP TO NEXT ERROR.
003426 000240 64$: NOP ;IF NO FORCE ERROR THEN JUST CONTINUE.
668 003430 032764 000200 000000 BT: BIT #BIT7,CSR(R4) ;DID INTR ON 'B' OCCUR?
669
670 003436 001676 BEQ TDAT ;IF NOT SEND MORE DATA
671 003440 WAIT,1 50,DATAACC,CIR,LB ;CHECK LAST DATA BYTE
003440 012767 000050 002414 MOV #50,DELCNT ; WAIT ( 50 US ) FOR DATAACC = 1.
003446 032765 100000 000000 64$: BIT @DATAACC,CIR(R5)
; BIT DATAACC = 1?
; YES.
; NO. BREAK TO MONITOR.
003454 001032 BNE LB ;TEMPORARY RETURN TO MONITOR...
003456 BREAK ;THEN CONTINUE AT NEXT INSTRUCTION.
003456 104407 000000' DEC DELCNT ; TIMEOUT?
003462 104407 000000' BREAK$,BEGIN ; NO.
003466 005367 002370 BREAK$,BEGIN ; YES.
003472 001365 BNE 64$ ;NO DATA ACCEPTED
672 003474 ER18: ERRB 66,25 ; SET UP FOR ERROR CALL
003474 004767 001252 JSR PC,ESETB ; SET ERROR CODE TYPE
003500 012767 000025 174400 MOV #25,ERRTYP ; SET ERROR CODE TYPE
003506 MSGN MSG66 ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
003506 104403 000000' 006220' MSGN$,BEGIN,MSG66 ;ASCII MESSAGE CALL WITH COMMON HEADER
003514 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
003514 104405 000000' 006110' HRDR$,BEGIN,REGTAB ;

```

```

;*****
003522 000240 NOP
003524 000240 NOP
673 003526 TEST FE,ER19
003526 005767 002350 TST FE ;TEST FOR 'FORCE ERROR' SET.
003532 001402 BEQ 64$ ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
003534 000167 000036 JMP ER19 ;JUMP TO NEXT ERROR.
003540 000240 64$: NOP ;IF NO FORCE ERROR THEN JUST CONTINUE.
674 003542 116522 000005 LB: MOVB IORHB(R5),(R2)+ ;READ LAST BYTE
675 003546 COMPARE TB5
003546 012703 000400 MOV #400,R3 ;SET UP FOR 256. COMPARES
003552 BUF1
003552 016701 174346 MOV RBUFVA,R1
003556 BUF2
003556 016702 174342 MOV RBUFVA,R2
003562 062702 000400 ADD #256.,R2
003566 005303 64$: DEC R3 ;KEEP TRACK
003570 001425 BEQ TB5 ;IF DONE GO TO NEXT TEST
003572 122221 CMPB (R2)+,(R1)+ ;COMPARE DATA
003574 001774 BEQ 64$ ; IF GOOD DO IT AGAIN
676 003576 ER19: ERRB 72,23 ;DATA IS NOT THE SAME
003576 004767 001150 JSR PC,ESETB ; SET UP FOR ERROR CALL
003602 012767 000023 174276 MOV #23,ERRTYP ; SET ERROR CODE TYPE
003610 MSGN MSG72 ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
003610 104403 000000' 006250' MSGN$,BEGIN,MSG72 ;ASCII MESSAGE CALL WITH COMMON HEADER
003616 HRDR REGTAB ; CALL ERROR. PRINT ALL REGS.
003616 104405 000000' 006110' HRDR$,BEGIN,REGTAB ;
;*****
003624 000240 NOP
003626 000240 NOP
677 003630 TEST FE,ER20
003630 005767 002246 TST FE ;TEST FOR 'FORCE ERROR' SET.
003634 001402 BEQ 64$ ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
003636 000167 000156 JMP ER20 ;JUMP TO NEXT ERROR.
678 003642 000240 64$: NOP ;IF NO FORCE ERROR THEN JUST CONTINUE.
679
680
681
682 ;**
683 ; IEC11-C TEST IEC11-A IS TALKER AND IEC11-B IS LISTENER
684 ;
685 ; THIS TEST WILL TRANSFER 256 BYTES FROM IEC11-A TO IEC11-B
686 ;*****
687 003644 TB5: MASTERCLEAR ;CLEAR THE WORLD
003644 052765 000040 000000 BIS #BIT5,CIR(R5)
003652 052764 000010 000000 BIS #BIT3,CSR(R4)
688
689 003660 MOV #400,R3 ;SET UP FOR 256 BYTE XFER
690 003664 BUF2
003664 016702 174234 MOV RBUFVA,R2
003670 062702 000400 ADD #256.,R2
691 003674 110322 BU: MOVB R3,(R2)+ ;PUT BYTES IN BUFFER
692 003676 005303 DEC R3 ;IS BUFFER FULL?
693 003700 001375 BNE BU ;NO GO STUFF ANOTHER BYTE
694 003702 SET.SACS ;SET SACS

```

```

695 003702 052765 000001 000000      BIS      #SACS,CIR(R5)
      003710 052765 000100 000002      SET.SIC      ;SET SIC
696 003716 012767 000050 002136      DELAY      50      ;GIVE IT SOME TIME.
      003724 012767 000050 002136      MOV        #50,DELCNT      ; DELAY ( 50 US ).
      003724 104407 000000' 64$:   BREAK      ; BREAK TO MONITOR.
      003730 104407 000000'      BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR....
      003734 005367 002122      BREAK$,BEGIN      ;THEN CONTINUE AT NEXT INSTRUCTION.
      003740 001371      DEC        DELCNT      ; TIMEOUT?
      003740 001371      BNE        64$      ; NO.
      003742 042765 177400 000000 G2:   BIC        #HIBYTE,CIR(R5)      ; YES.
      003742 042765 177400 000000      ;CLEAR INTERRUPTABLE BITS
698
699
700 003750 016700 002122      MOV        TBAS,RO      ;SET TALKER BASE ADDRESS
701 003754 066700 002126      ADD        AADR,RO      ;ADD IN IEC11A ADDRESS
702
703 003760 110065 000004      MOVB      RO,IOR(R5)      ;SEND IT DOWN THE BUS
704 003764 012767 000050 002070      WAIT.1    50,DATAACC,CIR,B$      ; IS DATA ACCEPTED?
      003764 012767 000050 002070      MOV        #50,DELCNT      ; WAIT ( 50 US ) FOR DATAACC = 1.
      003772 032765 100000 000000 64$:   BIT        #DATAACC,CIR(R5)
      004000 001032      BNE        B$      ; BIT DATAACC = 1?
      004002      BREAK      ; YES.
      004002 104407 000000'      BREAK$,BEGIN      ; NO. BREAK TO MONITOR.
      004006 104407 000000'      BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR....
      004012 005367 002044      DEC        DELCNT      ;THEN CONTINUE AT NEXT INSTRUCTION.
      004016 001365      BNE        64$      ; TIMEOUT?
      004020 001032      BNE        B$      ; NO.
      004020 004767 000726      ERRB      66,25      ; YES.
      004020 004767 000726      JSR        PC,ESETB      ;NO DATA ACCEPTED
      004024 012767 000025 174054      MOV        #25,ERRTYP      ; SET UP FOR ERROR CALL
      004032      MSGN      MSG66      ; SET ERROR CODE TYPE
      004032 104403 000000' 006220'      MSGN$,BEGIN,MSG66      ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
      004040      HRDR      REGTAB      ;ASCII MESSAGE CALL WITH COMMON HEADER
      004040 104405 000000' 006110'      HRDR$,BEGIN,REGTAB      ; CALL ERROR, PRINT ALL REGS.
      004046 000240      NOP      ;*****
      004050 000240      NOP      ;*****
706 004052      TEST      FE,ER21      ;TEST FOR 'FORCE ERROR' SET.
      004056 001402      BEQ        64$      ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
      004060 000167 000060      JMP        ER21      ; JUMP TO NEXT ERROR.
      004064 000240      NOP      ; IF NO FORCE ERROR THEN JUST CONTINUE.
707 004066 042765 100000 000000 64$:   CLR.DATAACC      ;RESET
      004066 042765 100000 000000 B$:   BIC        #DATAACC,CIR(R5)
708
709 004074 016700 002000      MOV        LBAS,RO      ;BUILD
710 004100 066700 002000      ADD        BADR,RO      ;LISTENER
711      ;ADDRESS
712      ;FOR
713      ;
714 004104 110065 000004      MOVB      RO,IOR(R5)      ;IEC11-B
715
716
717 004110      WAIT.1    50,DATAACC,CIR,C$

```

```

004110 012767 000050 001744      MOV    #50,DELCNT      ; WAIT ( 50 US ) FOR DATAACC = 1.
004116 032765 100000 000000 64$: BIT    #DATAACC,CIR(R5)
                                ; BIT DATAACC = 1?
                                ; YES.
004124 001032      BNE    C$             ; NO. BREAK TO MONITOR.
004126      BREAK                               ; TEMPORARY RETURN TO MONITOR....
004126 104407 000000'      BREAK$,BEGIN          ; THEN CONTINUE AT NEXT INSTRUCTION.
004132 104407 000000'      BREAK$,BEGIN
004136 005367 001720      DEC    DELCNT        ; TIMEOUT?
004142 001365      BNE    64$           ; NO.
                                ; YES.
718 004144      ER21:  ERRB   66,2$          ; NO DATA ACCEPTED
004144 004767 000602      JSR    PC,ESETB       ; SET UP FOR ERROR CALL
004150 012767 000025 173730      MOV    #25,ERRTP     ; SET ERROR CODE TYPE
004156      MSGN   MSG66          ; X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
004156 104403 000000' 006220'      MSGN$,BEGIN,MSG66    ; ASCII MESSAGE CALL WITH COMMON HEADER
004164      HRDR   REGTAB        ; CALL ERROR, PRINT ALL REGS.
                                ;*****
004164 104405 000000' 006110'      HRDR$,BEGIN,REGTAB  ;
                                ;*****
004172 000240      NOP
004174 000240      NOP
719 004176      TEST   FE,ER22          ; TEST FOR 'FORCE ERROR' SET.
004176 005767 001700      TST    FE             ; IF FORCE ERROR SET GO TO NEXT ERROR MSG.
004202 001402      BEQ    64$           ; JUMP TO NEXT ERROR.
004204 000167 000136      JMP    ER22
004210 000240      64$:  NOP           ; IF NO FORCE ERROR THEN JUST CONTINUE.
720 004212      C$:    CLR,DATAACC      ; RESET
004212 042765 100000 000000      BIC    #DATAACC,CIR(R5)
721
722
723 004220      SET,GTS          ; GO TO STANDBY
004220 052765 000004 000002      BIS    #GTS,SMR(R5)
724 004226 042764 177400 000000      BIC    #HIBYTE,CSR(R4) ; CLEAR INTERRUPTABLE BITS
725 004234      BUF1
004234 016701 173664      MOV    RBUFVA,R1
726 004240 016764 173662 000004      MOV    RBUFPA,BAR(R4) ; SET BUFFER FOR IEC11-B
727 004246 012703 000377      MOV    #377,R3        ; 256 BYTES WORTH
728 004252 005403      NEG    R3
729 004254 010364 000002      MCV    R3,BCR(R4)     ; SET UP BYTE COUNTER
730 004260      BUF2
004260 016702 173640      MOV    RBUFVA,R2
004264 062702 000400      ADD    #256,,R2
731
732 004270      GO          ; START TALKING
004270 056764 173634 000000      BIS    RBUFEA,CSR(R4)
004276 052764 000001 000000      BIS    #1,CSR(R4)
733
734 004304 005003      CLR    R3
735 004306 112265 000004      TALK:  MOVB   (R2)+,IOR(R5) ; I'M TALKING
736 004312      WAIT,1 50,DATAACC,CIR,D2 ; DID YOU HEAR ME?
004312 012767 000050 001542      MOV    #50,DELCNT    ; WAIT ( 50 US ) FOR DATAACC = 1.
004320 032765 100000 000000 64$: BIT    #DATAACC,CIR(R5)
                                ; BIT DATAACC = 1?
                                ; YES.
004326 001032      BNE    D2             ; NO. BREAK TO MONITOR.
004330      BREAK                               ; TEMPORARY RETURN TO MONITOR....
004330 104407 000000'      BREAK$,BEGIN          ; THEN CONTINUE AT NEXT INSTRUCTION.
004334 104407 000000'      BREAK$,BEGIN

```

```

004340 005367 001516      DEC      DELCNT      ; TIMEOUT?
004344 001365      BNE      64$        ; NO.
                                ; YES.
737 004346      ER22:  ERRB      66,25      ;NO DATA ACCEPTED
004346 004767 000400      JSR      PC,ESETB   ; SET UP FOR ERROR CALL
004352 012767 000025 173526  MOV      #25,ERRTYP ; SET ERROR CODE TYPE
004360      MSGN      MSG66      ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
004360 104403 000000' 006220'  MSGN$,BEGIN,MSG66 ;ASCII MESSAGE CALL WITH COMMON HEADER
004366      HRDR      REGTAB      ; CALL ERROR, PRINT ALL REGS.
                                ;*****
004366 104405 000000' 006110'  HRDR$,BEGIN,REGTAB ;
                                ;*****
004374 000240      NOP
004376 000240      NOP
738 004400      TEST      FE,ER23
004400 005767 001476      TST      FE          ;TEST FOR 'FORCE ERROR' SET.
004404 001402      BEQ      64$        ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
004406 000167 000030      JMP      :R23       ;JUMP TO NEXT ERROR.
004412 000240      NOP          ;IF NO FORCE ERROR THEN JUST CONTINUE.
739 004414      64$:  CLR.DAT,ACC      ;YES=RESET
004414 042765 100000 000000  D2:  BIC      #DATAACC,CIR(R5)
740 004422 032764 000200 000000  BIT      #BIT7,CSR(R4)      ;LAST BYTE?
741 004430 001027      BNE      CKDT5      ;YES=GO CHECK DATA
742 004432 005203      INC      R3          ;COUNT BYTES ANYWAY
743 004434 022703 000400      CMP      #400,R3     ;SHOULD NEVER GET HERE
744 004440 001322      BNE      TALK       ;I'VE GOT MORE TO SAY
745 004442      ER23:  ERRB      71,23      ;BYTE COUNTER NEVER OVFL'D
004442 004767 000304      JSR      PC,ESETB   ; SET UP FOR ERROR CALL
004446 012767 000023 173432  MOV      #23,ERRTYP ; SET ERROR CODE TYPE
004454      MSGN      MSG71      ;X-REF ERROR CODE & DESCRIPTIVE MESSAGE.
004454 104403 000000' 006244'  MSGN$,BEGIN,MSG71 ;ASCII MESSAGE CALL WITH COMMON HEADER
004462      HRDR      REGTAB      ; CALL ERROR, PRINT ALL REGS.
                                ;*****
004462 104405 000000' 006110'  HRDR$,BEGIN,REGTAB ;
                                ;*****
004470 000240      NOP
004472 000240      NOP
746 004474      TEST      FE,ER24
004474 005767 001402      TST      FE          ;TEST FOR 'FORCE ERROR' SET.
004500 001402      BEQ      64$        ;IF FORCE ERROR SET GO TO NEXT ERROR MSG.
004502 000167 000032      JMP      ER24       ;JUMP TO NEXT ERROR.
004506 000240      NOP          ;IF NO FORCE ERROR THEN JUST CONTINUE.
747
748 004510      CKDT5:  COMPARE  TEND
004510 012703 000400      MOV      #400,R3     ;SET UP FOR 256 COMPARES
004514      BUF1
004514 016701 173404      MOV      RBUFVA,R1
004520      BUF2
004520 016702 173400      MOV      RBUFVA,R2
004524 062702 000400      ADD      #256.,R2
004530      64$:  DEC      R3          ;KEEP TRACK
004532 001425      BEQ      TEND       ;IF DONE GO TO NEXT TEST
004534 122221      CMPB     (R2)+,(R1)+ ;COMPARE DATA
004536 001774      BEQ      64$        ; IF GOOD DO IT AGAIN
749 004540      ER24:  ERRB      72,23      ;ERROR DATA NOT EQUAL
004540 004767 000206      JSR      PC,ESETB   ; SET UP FOR ERROR CALL
004544 012767 000023 173334  MOV      #23,ERRTYP ; SET ERROR CODE TYPE

```



```

757                                     .SBTTL  INTERRUPT SERVICE ROUTINE
758
759 004632                               INTSVC: PIRQ   GOIRTN
                                     |-----|
                                     | PIRQ$,BEGIN,GOIRTN           | QUEUE UP TO CONTINUE AT GOIRTN AND RTI
                                     |-----|
760
761 004640 016705 173230                 GOIRTN: MOV     SVR5,R5           | RESTORE R5.
762 004644 142765 177400 000000        BICB   @HI BYTE,CIR(R5)       | CLEAR INTERRUPT BITS.
763 004652 000177 001212                JMP     @IRTN                 | RETURN.
764
765 004656                               INTSVB: PIRQ   ITRB
                                     |-----|
                                     | PIRQ$,BEGIN,ITRB           | QUEUE UP TO CONTINUE AT ITRB AND RTI
                                     |-----|
766
767 004664 016704 173202                 ITRB:  MOV     SVR4,R4           | RESTORE R4.
768 004670 142764 177400 000000        BICB   @HI BYTE,CSR(R4)     | CLEAR INT BITS.
769 004676 000177 001156                JMP     @MYRTN                | RETURN.
770
771
772
773
774
775
776
777

```

```

780          .SBTTL  SUBROUTINES
781
782          ;;;
783          ; SUBROUTINE - SET UP FOR ERROR CALL.
784          ;--
785
786 004702 010567 173172      ESETUP: MOV      R5,CSRA          ; SET UP CSR (CIR) ADDRESS.
787 004706 011567 173170      MOV      (R5),ACSR        ; SET UP CSR (CIR) CONTENTS.
788 004712 010500              MOV      R5,R0          ; SET UP REG POINTERS IN TABLE.
789 004714 010067 001170      MOV      R0,CIRPTR
790 004720 062700 000002      ADD      @2,R0
791 004724 010067 001162      MOV      R0,SMRPTR
792 004730 062700 000002      ADD      @2,R0
793 004734 010067 001154      MOV      R0,IORPTR
794 004740 062700 000002      ADD      @2,R0
795 004744 010067 001146      MOV      R0,VSRPTR
796 004750 000207              RTS      PC
797
798
799 004752 010467 173122      ESETB:  MOV      R4,CSRA          ; SET UP CSR ADDRESS
800 004756 011467 173120      MOV      (R4),ACSR        ; SET UP CSR CONTENTS
801 004762 010400              MOV      R4,R0          ; SET UP REG POINTERS IN TABLE
802 004764 010067 001130      MOV      R0,CSRPTR
803 004770 062700 000002      ADD      @2,R0
804 004774 010067 001122      MOV      R0,BCRPTR
805 005000 062700 000002      ADD      @2,R0
806 005004 010067 001114      MOV      R0,BARPTR
807 005010 062700 000002      ADD      @2,R0
808 005014 010067 001106      MOV      R0,MCRPTR
809 005020 010500              MOV      R5,R0          ; SET UP FOR IEC11-A
810 005022 010067 001062      MOV      R0,CIRPTR
811 005026 062700 000002      ADD      @2,R0
812 005032 010067 001054      MOV      R0,SMRPTR
813 005036 062700 000002      ADD      @2,R0
814 005042 010067 001046      MOV      R0,IORPTR
815 005046 062700 000002      ADD      @2,R0
816 005052 010067 001040      MOV      R0,VSRPTR
817 005056 000207              RTS      PC

```

820			.SBTTL	CONSTANTS AND VARIABLES.
821	005060		BUFIN: .BLKB	512.
822	006060	000000	MYRTN: 0	; RETURN ADDRESS FOR INTERRUPTS
823	006062	000000	DELCNT: 0	; DELAY COUNTER.
824	006064	000000	RANMOD: 0	; RANDOM MODE (0=NOT RANDOM, -1=RANDOM)
825	006066	000000	LOOP: 0	; LOOP COUNTER.
826	006070	000000	IRTN: 0	; INTERRUPT RETURN ADDRESS.
827	006072	000000	DATBUF: 0	; DATA WORD OR BYTE BUFFER.
828	006074	000000	GOT: 0	; DATA ACTUALLY GOTTEN.
829	006076	000100	TBAS: 100	; TALKER BASE ADDRESS
830	006100	000040	LBAS: 40	; LISTENER BASE ADDRESS
831	006102	000000	FE: 0	; FORCE ERROR -NORMALLY ZERO -WHEN SET WILL FORCE ALL ERROR PRINTOUTS
832	006104	000036	BADR: 36	; IEC11-B IEEE BUS ADDRESS
833	006106	000035	AADR: 35	; IEC11-A IEEE BUS ADDRESS
834				
835	006110		REGTAB:	; REGISTER POINTERS SAVED HERE FOR ERR DUMPING.
836	006110	000000	CIRPTR: CIR	; CIR POINTER (AFTER R5 ADDED IN).
837	006112	000002	SMRPTR: SMR	; SMR POINTER (AFTER R5 ADDED IN).
838	006114	000004	IORPTR: IOR	; IOR POINTER (AFTER R5 ADDED IN).
839	006116	000006	VSRPTR: VSR	; VSR POINTER (AFTER R5 ADDED IN).
840	006120	000000	CSRPTR: CSR	; CSR POINTER (AFTER R4 ADDED IN.)
841	006122	000002	BCRPTR: BCR	; BCR POINTER
842	006124	000004	BARPTR: BAR	; BAR POINTER
843	006126	000006	MCRPTR: MCR	; MCR POINTER
844	006130	006072	DBPTR: DATBUF	; DATBUF (DATA BUFFER) POINTER.
845	006132	177777	-1	; TABLE TERMINATOR.

851			.SBTTL	ASCII MESSAGES
852				
853	006134	006254'	MSG46:	EMSG46
854	006136	177777		-1
855	006140	006330'	MSG50:	EMSG50
856	006142	177777		-1
857	006144	006431'	MSG52:	EMSG52
858	006146	177777		-1
859	006150	006532'	MSG53:	EMSG53
860	006152	177777		-1
861	006154	006606'	MSG54:	EMSG54
862	006156	177777		-1
863	006160	006743'	MSG55:	EMSG55
864	006162	177777		-1
865	006164	007040'	MSG56:	EMSG56
866	006166	177777		-1
867	006170	007173'	MSG57:	EMSG57
868	006172	177777		-1
869	006174	007271'	MSG60:	EMSG60
870	006176	177777		-1
871	006200	007366'	MSG62:	EMSG62
872	006202	177777		-1
873	006204	007454'	MSG63:	EMSG63
874	006206	177777		-1
875	006210	007550'	MSG64:	EMSG64
876	006212	177777		-1
877	006214	007612'	MSG65:	EMSG65
878	006216	177777		-1
879	006220	007656'	MSG66:	EMSG66
880	006222	177777		-1
881	006224	007731'	MSG67:	EMSG67
882	006226	177777		-1
883	006230	010005'	MSG68:	EMSG68
884	006232	177777		-1
885	006234	010053'	MSG69:	EMSG69
886	006236	177777		-1
887	006240	010103'	MSG70:	EMSG70
888	006242	177777		-1
889	006244	010135'	MSG71:	EMSG71
890	006246	177777		-1
891	006250	010206'	MSG72:	EMSG72
892	006252	177777		-1
893				

894	006254	105	122	122	EMSG46:	.ASCIZ	'ERROR 46 - "TCS" DOES NOT PRODUCE "ILLMSGE"'
895	006330	105	122	122	EMSG50:	.ASCIZ	'ERROR 50 - WHEN SENDING THE LISTENER ADDRESS, NO INTERRUPT OCCURS'
896	006431	105	122	122	EMSG52:	.ASCIZ	'ERROR 52 - AFTER LISTENER ADDRESSING AND "GTS", NO "LACS" IS SET'
897	006532	105	122	122	EMSG53:	.ASCIZ	'ERROR 53 - NO INTERRUPT TO TALKER ADDRESSING'
898	006606	105	122	122	EMSG54:	.ASCII	'ERROR 54 - WHEN SENDING A SECOND DATA BYTE BEFORE READING'
899	006677	045	124	110		.ASCIZ	'*THE FIRST AGAIN, "DATAACC" CAME UP'
900	006743	105	122	122	EMSG55:	.ASCIZ	'ERROR 55 - FIRST DATA BYTE SENT AND RECEIVED IS NOT THE SAME'
901	007040	105	122	122	EMSG56:	.ASCII	'ERROR 56 - AFTER READING THE FIRST SENT DATA BYTE, THE SECOND SENT'
902	007142	045	104	111		.ASCIZ	'*DID NOT BRING "DATAACC"'
903	007173	105	122	122	EMSG57:	.ASCIZ	'ERROR 57 - SECOND DATA BYTE SENT AND RECEIVED IS NOT THE SAME'
904	007271	105	122	122	EMSG60:	.ASCIZ	'ERROR 60 - SENDING OF DATA BYTE DOES NOT PRODUCE AN INTERRUPT'
905	007366	105	122	122	EMSG62:	.ASCIZ	'ERROR 62 - DATA BYTES SENT AND RECEIVED ARE NOT EQUAL'
906	007454	105	122	122	EMSG63:	.ASCIZ	'ERROR 63 - NO "DATAACC" WHEN SENDING A BYTE WITH "LASTBYTE"'
907	007550	105	122	122	EMSG64:	.ASCIZ	'ERROR 64 - NO "END" TO "LASTBYTE"'

ASCII MESSAGES

908	007612	105	122	122	EMSG65: .ASCIZ	'ERROR 65 - DATA CHECK ON "LASTBYTE"'
909	007656	105	122	122	EMSG66: .ASCIZ	'ERROR B45- NO DATA ACCEPTED (IEC11-B TEST)'
910	007731	105	122	122	EMSG67: .ASCIZ	'ERROR B46- NOT ENOUGH BYTES TRANSFERRED YET'
911	010005	105	122	122	EMSG68: .ASCIZ	'ERROR B47- TOO MANY BYTES TRANSFERRED'
912	010053	105	122	122	EMSG69: .ASCIZ	'ERROR B48- CACS NOT SET'
913	010103	105	122	122	EMSG70: .ASCIZ	'ERROR B49- SIAS NOT CLEAR'
914	010135	105	122	122	EMSG71: .ASCIZ	'ERROR B50- BYTE COUNTER NEVER OVERFLOWED'
915	010206	105	122	122	EMSG72: .ASCIZ	'ERROR B51- DATA SENT NOT EQUAL TO DATA RECD'
916						

H3
IECA DEC/X11 SYSTEM EXERCISER M MACRO M1200 30-MAR-84 11:10 PAGE 23
END OF PROGRAM

SEQ 33

922
923 010262 000001

PEND: .SBTTL END OF PROGRAM
.END

AADR	006106R	C1	003002R	ER7	001606R	MSG54	006154R	RESTR	000252R
ACSR	000102R	C2	003126R	ER8	001702R	MSG55	006160R	RES1	000056R
ADDR	000006R	DATAAC	100000	ER9	002040R	MSG56	006164R	RES2	000060R
ADDR22	001000	DATBUF	006072R	ESETB	004752R	MSG57	006170R	RH70	001000
ALL	177777	DATCK	104411	ESETUP	004702R	MSG60	006174R	RSET	003340R
APTPRE	000200	DAYER	104404	EXIT	104400	MSG62	006200R	RSTR	000112R
ASB	000106R	DBPTR	006130R	FE	006102R	MSG63	006204R	R6	000006
ASTAT	000104R	DELCNT	006062R	FREE	000150R	MSG64	006210R	R7	000007
AUTO	000010	DVID1	000014R	GETPA	104415	MSG65	006214R	SACS	000001
AWAS	000110R	D2	004414R	GOIRTN	004640R	MSG66	006220R	SBADR	000102R
BADR	006104R	ECCMEM	000100	GOT	006074R	MSG67	006224R	SIAS	020000
BAR	000004	EF	000001	GTS	000004	MSG68	006230R	SIC	000100
BARPTR	006124R	EMSG46	006254R	GWBUF	104414	MSG69	006234R	SMR	000002
BCR	000002	EMSG50	006330R	G2	003742R	MSG70	006240R	SMRPTR	006112R
BCRPTR	006122R	EMSG52	006431R	HIBYTE	177400	MSG71	006244R	SOFCNT	000042R
BEGIN	000000R	EMSG53	006532R	HRDCNT	000044R	MSG72	006250R	SOFR	104406
BIT0	000001	EMSG54	006606R	HRDER	104405	MYRTN	006060R	SOFPAS	000046R
BIT1	000002	EMSG55	006743R	HRDPAS	000050R	NCPUPP	000020	SPOINT	000032R
BIT10	002000	EMSG56	007040R	ICONT	000036R	NOPTY	000002	SPSIZ	000040
BIT11	004000	EMSG57	007173R	ICOUNT	000040R	NULL	000000	SR1	000016R
BIT12	010000	EMSG60	007271R	IDNUM	000122R	OPEN	000000	SR2	000020R
BIT13	020000	EMSG62	007366R	IE	000100	OTOA	104420	SR3	000022R
BIT14	040000	EMSG63	007454R	ILLMSG	010000	PARPRE	002000	SR4	000024R
BIT15	100000	EMSG64	007550R	IMODX	000000	PASCNT	000034R	START	000252R
BIT2	000004	EMSG65	007612R	INDPAR	000040	PDPF11	000002	STAT	000026R
BIT3	000010	EMSG66	007656R	INIT	000030R	PDPLSI	020000	STATEC	000400
BIT4	000020	EMSG67	007731R	INTENB	000100	PDP44	100000	SVR0	000062R
BIT5	000040	EMSG68	010005R	INTR	000120R	PDP60	004000	SVR1	000064R
BIT6	000100	EMSG69	010053R	INTSVB	004656R	PDP70	010000	SVR2	000066R
BIT7	000200	EMSG70	010103R	INTSVC	004632R	PEND	010262R	SVR3	000070R
BIT8	000400	EMSG71	010135R	IOR	000004	PIRQ	000004	SVR4	000072R
BIT9	001000	EMSG72	010206R	IORMB	000005	POPSP	005726	SVR5	000074R
BREAK	104407	END	020000	IORPTR	006114R	POPSP2	022626	SVR6	000076R
BR1	000012R	ENDIT	104413	IRTN	006070R	PRHMS	000002	SYSCNT	000052R
BR2	000013R	END	104410	ITRB	004664R	PRTY	000000	TALK	004306R
BT	003430R	ERRTYP	000106R	KTPRES	000400	PRTY0	000000	TBAS	006076R
BTOD	104421	ER1	000650R	KTXTND	040000	PRTY1	000040	TB2	002656R
BU	003674R	ER10	002134R	LACS	010000	PRTY2	000100	TB5	003644R
BUFIN	005060R	ER11	002342R	LASTBY	000004	PRTY3	000140	TCA	000002
B	004066R	ER12	002420R	LB	003542R	PRTY4	000200	TCS	000001
CACS	000400	ER13	002514R	LBAS	006100R	PRTY5	000240	TDAT	003234R
CAPRES	000004	ER14	002734R	LOBYTE	000377	PRTY6	000300	TEND	004606R
CDATA	104412	ER15	003060R	LOOP	006066R	PRTY7	000340	TRPDFD	000023
CDERCT	000144R	ER16	003272R	MAP22	104416	PS	177776	TSTIEA	000426R
CDWDCT	000146R	ER17	003362R	MC	000040	PSW	177776	TST4B	002562R
CIR	000000	ER18	003474R	MCB	000010	PUSH	005746	T1	000442R
CIRPTR	006110R	ER19	003576R	MCR	000006	PUSH2	024646	T10	001750R
CKDT5	004510R	ER2	001030R	MCRPTR	006126R	PWRFLG	000002	T11	002006R
CKHNG	000001	ER20	004020R	MODNAM	000000R	QMON22	000010	T12	002106R
CLKPRE	000001	ER21	004144R	MODSP	000252R	RAND	104417	T13	002106R
CLKSP	104422	ER22	004346R	MSGN	104403	RANMOD	006064R	T14	002202R
CONF IG	000056R	ER23	004442R	MSG5	104402	RANNUM	000054R	T15	002232R
CSBS	001000	ER24	004540R	MSG	104401	RBUFEA	000130R	T16	002410R
CSR	000000	ER3	001150R	MSG46	006134R	RBUFPA	000126R	T17	002466R
CSRA	000100R	ER4	001260R	MSG50	006140R	RBUFSZ	000132R	T2	000716R
CSRPTR	006120R	ER5	001434R	MSG52	006144R	RBUFVA	000124R	T3	001076R
C	004212R	ER6	001530R	MSG53	006150R	REGTAB	006110R	T4	001076R

J3

T5	001216R	T9	001654R	VSRPTR	006116R	WBUFPA	000134R	WDFR	000116R
T6	001326R	USTACK=	000001	WASADR	000104R	WBUFQ	000140R	WDTO	000114R
T7	001502R	VECTOR	000010R	WBUFEA	000136R	WBUFSZ	000142R	XFLAG	000005R
T8	001576R	VSR	= 000006						

. ABS. 000000 000
010262 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 15112 WORDS (60 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:00:38
XIECAO,XIECAO/-SP=DDXCOM.MAC,XIECAO.MAC

DOCUMENTB1
DOCUMENTC1
DOCUMENTD1
DOCUMENTE1
DDXCOM HEADERF1
DDXCOM HEADERG1
MACRO DEFINITIONS.H1
MACRO DEFINITIONS.I1
MACRO DEFINITIONS.J1
EQUATESK1
START - RESTART - TE....L1
START - RESTART - TE....M1
START - RESTART - TE....N1

START - RESTART - TE....B2
START - RESTART - TE....C2
START - RESTART - TE....D2
START - RESTART - TE....E2
START - RESTART - TE....F2
START - RESTART - TE....G2
START - RESTART - TE....H2
START - RESTART - TE....I2
START - RESTART - TE....J2
START - RESTART - TE....K2
START - RESTART - TE....L2
START - RESTART - TE....M2
START - RESTART - TE....N2

START - RESTART - TE....B3
INTERUPT SERVICE ROU....C3
SUBROUTINESD3
CONSTANTS AND VARIAB....E3
ASCII MESSAGESF3
ASCII MESSAGESG3
END OF PROGRAMH3
SYMBOL TABLEI3
SYMBOL TABLEJ3