

.REM\_

IDENTIFICATION  
-----

PRODUCT CODE: AC-T885A-MC  
PRODUCT TITLE: CXIEBA0 IEC11-A DEC/X11 MODULE  
PRODUCT DATE: JANUARY 1984  
DEPARTMENT: CSS, NASHUA  
AUTHOR: DON MACOMBER

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 TRANSFERED.  
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.

DOCUMENT

1 A B S T R A C T

IEB IS AN IOMOD WHICH EXERCISES THE IEC11-A IEC BUS CONTROLLER. IT EXERCISES THE IEC11-A BY DOING DATA TRANSFERS WITH INTERRUPT, AND BY DOING A RANDOM TALKER AND LISTENER DATA TEST. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2 R E Q U I R E M E N T S

HARDWARE:

- A. PDP11 CAPABLE OF SUPPORTING DEC/X11.
- B. IEC11-A IEC BUS CONTROLLER (M7985).
- C. ANY USER IEC BUS DEVICES MUST BE DISCONNECTED.
- D. THE IEC BUS DEVICE ADDRESS MUST BE SET TO 35(8).

SOFTWARE:

DEC/X11 MONITOR, VERSION 6.

STORAGE:

- 1. DECIMAL WORDS: 1016
- 2. OCTAL WORDS: 1770
- 3. OCTAL BYTES: 3760

3 P A S S D E F I N I T I O N

TEN PASS OF THE IEB MODULE CONSISTS OF TEN CYCLES OF THE TEST.

4 E X E C U T I O N T I M E

ONE PASS OF IEB RUNNING ALONE ON A PDP-11/34 TAKES APPROXIMATELY 40 SECONDS.

5 C O N F I G U R A T I O N R E Q U I R E M E N T S

DEFAULT PARAMETERS:

DEVADR: 160010, VECTOR: 270, BR1: 6, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

DOCUMENT

```

6  DEVICE / OPTION  SETUP
   *****
   MAKE CERTAIN THAT THE IEC11-A IS CONNECTED AND READY.

7  MODULE OPERATION
   *****
   TEST SEQUENCE:
   A. TEST DATA TRANSFER WITH INTERUPT, CHECKING THAT:
      1. 'TCS' PRODUCES 'ILLMSGE'
      2. INTERUPT OCCURS WHEN SENDING LISTENER ADDRESS
      3. 'LACS' IS SET AFTER LISTENER ADDRESSING AND 'GTS'
      4. INTERUPT 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. INTERUPT 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

8  OPERATION OPTIONS
   *****
   NCNE

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 (CZIEBAO).
   THE SECOND SECTION CONSISTS OF A STANDARD DEC/X11
   HARD ERROR (HRDFP) 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

```

E1

- 10    S T A R T I N G   P R O C E D U R E  
      \*\*\*\*\*
- A.    LOAD AND START THE DEC/X11 PROGRAM ACCORDING TO  
            THE DEC/X11 USER'S MANUAL (AC-82340-MC).
- B.    ENSURE THAT THE DEVICE ADDRESS AND VECTOR ADDRESS  
            ARE CORRECT FOR THE IEC11-A BUS CONTROLLER.
  
- 11    L I S T I N G  
      \*\*\*\*\*

179  
180  
181 000000  
000000

.SHTTL DDXCOM HEADER

IOMOD <IEBA >,160010,270,6,0,0,10,,0  
MODULE 140000,IEBA ,160010,270,6,0,0,10,,0,,,,  
.TITLE IEBA DEC/X11 SYSTEM EXERCISER MODULE  
; DDXCOM VERSION 6.4 28-JAN-82  
.LIST BIN

\*\*\*\*\*

000000  
000000 111 105  
000003 101 040  
000005 000  
000006 160010  
000010 000270  
000012 300  
000013 000  
000014 000001  
000016 000000  
000020 000000  
000022 000000  
000024 000000

102

BEGIN:  
MODNAM: .ASCII /IEBA / ;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 PRTY0+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

\*\*\*\*\*

000026 140000  
000030 000224  
000032 000224  
000034 000000  
000036 000012  
000040 000000  
000042 000000  
000044 000000  
000046 000000  
000050 000000  
000052 000000  
000054 000000  
000056 000000  
000060 000000  
000062 000000  
000064 000000  
000066 000000  
000070 000000  
000072 000000  
000074 000000  
000076 000000  
000100 000000  
000102 000000  
000104 000000  
000104 000000  
000106 000000  
000106 000000  
000110 000000  
000112 000224  
000114 000000  
000116 000000  
000120 000000  
000122 000000

STAT: 140000 ;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  
HADCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS  
SOPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS  
HADPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS  
SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED  
RANUM: 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.  
RSTR: RERSTRT ;RESTART ADDRESS AFTER END OF PASS  
WDTO: OPEN ;WORDS TO MEMORY PER ITERATION  
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION  
INTR: OPEN ;# OF INTERRUPTS PER ITERATION  
IDNUM: 0 ;MODULE IDENTIFICATION NUMBER=0

G1

C00040

.REPT SPSIZ  
.NLIST  
.WORD 0  
.LIST  
.ENDR

;MODULE STACK STARTS HERE.

000224

MODSP:

\*\*\*\*\*

## MACRO DEFINITIONS.

```

184      .SBTTL  MACRO DEFINITIONS.
185
186      .MACRO  SET,SACS
187      BIS     @SACS,CIR(R5)
188      .ENDM   SET,SACS
189
190      .MACRO  SET,LASTBYTE
191      BIS     @LASTBYTE,CIR(R5)
192      .ENDM   SET,LASTBYTE
193
194      .MACRO  SET,MC
195      BIS     @MC,CIR(R5)
196      .ENDM   SET,MC
197
198      .MACRO  SET,INTENB
199      BIS     @INTENB,CIR(R5)
200      .ENDM   SET,INTENB
201
202      .MACRO  SET,CA
203      BIS     @TCA,SMR(R5)
204      .ENDM   SET,CA
205
206      .MACRO  SET,CS
207      BIS     @TCS,SMR(R5)
208      .ENDM   SET,CS
209
210      .MACRO  SET,GTS
211      BIS     @GTS,SMR(R5)
212      .ENDM   SET,GTS
213
214      .MACRO  SET,SIC
215      BIS     @SIC,SMR(R5)
216      .ENDM   SET,SIC
217
218      .MACRO  CLR,STATECHGE
219      BIC     @STATECHGE,CIR(R5)
220      .ENDM   CLR,STATECHGE
221
222      .MACRO  CLR,DATAACC
223      BIC     @DATAACC,CIR(R5)
224      .ENDM   CLR,DATAACC
225
226      .MACRO  TST,0  ABIT,REG,GOIF0
227      BIT     @ABIT,REG(R5)
228
229      BEQ     GOIF0                                ; ABIT CLEAR IN REG ?
230
231      .ENDM   TST,0                                ; YES.
232
233      .MACRO  TST,1  ABIT,REG,GOIF1
234      BIT     @ABIT,REG(R5)
235
236      BNE     GOIF1                                ; ABIT SET IN REG ?
237
238      .ENDM   TST,1                                ; YES.
239
240      .MACRO  DELAY  TIME,PLS

```

## MACRO DEFINITIONS.

```

241      MOV      @TIME,DELCNT          ; DELAY ( TIME US ).
242      L$:     BREAK                    ; BREAK TO MONITOR.
243      DEC      DELCNT                ; TIMEOUT?
244      BNE     L$                     ; NO.
245      ;                                     ; YES.
246      .ENDM   DELAY
247
248      .MACRO  WAIT.0  TIME,ABIT,REG,GOIFO,?L$
249      MOV      @TIME,DELCNT          ; WAIT ( TIME US ) FOR ABIT = 0.
250      L$:     BIT      @ABIT,REG(R5)
251      ;                                     ; BIT ABIT = 0?
252      BEQ     GOIFO                    ; YES.
253      BREAK                    ; NO. BREAK TO MONITOR.
254      DEC      DELCNT                ; TIMEOUT?
255      BNE     L$                     ; NO.
256      ;                                     ; YES.
257      .ENDM   WAIT.0
258
259      .MACRO  WAIT.1  TIME,ABIT,REG,GOIF1,?L$
260      MOV      @TIME,DELCNT          ; WAIT ( TIME US ) FOR ABIT = 1.
261      L$:     BIT      @ABIT,REG(R5)
262      ;                                     ; BIT ABIT = 1?
263      BNE     GOIF1                    ; YES.
264      BREAK                    ; NO. BREAK TO MONITOR.
265      DEC      DELCNT                ; TIMEOUT?
266      BNE     L$                     ; NO.
267      ;                                     ; YES.
268      .ENDM   WAIT.1
269
270      .MACRO  WAIT.INT
271      MOV      R5,SVR5                ; SAVE CONTENTS OF R5.
272      EXIT                    ; WAIT FOR INTERRUPT - EXIT TO MONITOR.
273
274      .ENDM   WAIT.INT
275
276      .MACRO  ERROR  XREF,ECODE
277      JSR     PC,ESETUP                ; SET UP FOR ERROR CALL.
278      MOV     @ECODE,ERRTYP            ; SET ERROR CODE TYPE.
279      MSGN   MSG'XREF'                ; X-REF ERR CODE & DESCRIPTIVE MSG.
280      HRDER  REGTAB                    ; CALL ERROR, PRINT ALL REGS.
281      NOP
282      NOP
283      .ENDM   ERROR
284
285      .MACRO  SCOPE
286      NOP
287      NOP
288      .ENDM   SCOPE

```



```

291          .SBTTL  EQUATES
292
293          ;**
294          ; DEVICE REGISTERS AND BIT DEFINITIONS.
295          ;**
296
297          000000  CIR=    0      ; CONTROL & INTERRUPT REGISTER.
298          000001  SACS=    BIT0    ; SYSTEM CONTROL ACTIVE STATE.
299          000004  LASTBYTE= BIT2    ; LAST BYTE.
300          000040  MC=      BIT5    ; MASTER CLEAR.
301          000100  INTENB=  BIT6    ; INTERRUPT ENABLE.
302          000400  STATECHG= BIT8    ; STATE CHANGE.
303          010000  ILLMSG=  BIT12   ; ILLEGAL MESSAGE.
304          020000  END=    BIT13   ; END OF BLOCK.
305          100000  DATAACC= BIT15  ; DATA ACCEPTED.
306
307          000002  SMR=    2      ; STATE & MESSAGE REGISTER.
308          000002  TCA=    BIT1    ; TAKE CONTROL ASYNCHRONOUSLY.
309          000001  TCS=    BIT0    ; TAKE CONTROL SYNCHRONOUSLY.
310          000004  GTS=    BIT2    ; GO TO STANDBY.
311          000100  SIC=    BIT6    ; SEND INTERFACE CLEAR.
312          000400  CACS=    BIT8    ; CONTROLLER ACTIVE STATE.
313          001000  CSBS=    BIT9    ; CONTROLLER STANDBY STATE.
314          010000  LACS=    BIT12   ; LISTENER ACTIVE STATE.
315          020000  SIAS=    BIT13   ; INTERFACE CLEAR ACTIVE STATE.
316
317          000004  IOR=    4      ; INPUT & OUTPUT REGISTER.
318
319          000006  VSR=    6      ; VECTOR SWITCH REGISTER.
320
321          ;**
322          ; GENERAL EQUATES.
323          ;**
324
325          177777  ALL=    177777  ; ALL BITS (BITS 15-0).
326          000377  LOBYTE= 377    ; LO ORDER BITS (BITS 7-0).
327          177400  HIBYTE= 177400 ; HI ORDER BITS (BITS 15-8).

```

```

330                                     .SBTTL  START - RESTART - TEST
331
335
336 000224      START:
337 000224      RESTART:
338
339      ;++
340      ; SET UP VECTOR, BR LEVEL, AND R5 AS DEVICE POINTER.
341      ;--
342
343 000224      016705      177560      MOV      VECTOR,R5      ; GET VECTOR ADDRESS.
344 000230      012725      002136      MOV      #INTSVC,(R5)+    ; SET COMPLETION INTERRUPT VECTOR.
345 000231      116715      177552      MOV      BR1,(R5)      ; SET BREAK LEVEL.
346 000240      016705      177542      MOV      #DDR,R5      ; R5 = DEVICE BASE ADDRESS.
347 000244      005067      001776      CLR      DATBUF      ; CLEAR THE DATA BUFFER.
348
349      ;++
350      ; TEST DATA TRANSFER WITH INTERRUPT.
351      ;--
352
353 000250      TSTIEB:      ; SET UP AND TEST THE IEC11-A.
354 000250      11:      ;
355 000250      052765      000040      000000      SET.MC      ; MASTER CLEAR.
356 000256      052765      000001      000000      BIS      #MC,CIR(R5)      ; SYSTEM CONTROL ACTIVE STATE.
357 000264      052765      000100      000002      SET.SACS      ; SEND INTERFACE CLEAR.
358 000272      052765      000100      000002      BIS      #SACS,CIR(R5)
359 000272      012767      000100      001732      SET.SIC      ; (GIVE STATECHGE TIME TO CLEAR).
360 000272      012767      000100      001732      DELAY      100      ; DELAY ( 100 US ).
361 000300      104407      000000      64$:      MOV      #100,LE CNT      ; BREAK TO MONITOR.
362 000300      104407      000000      BREAK      ; TEMPORARY RETURN TO MONITOR....
363 000304      005367      001716      BREAK$,BEGIN      ; THEN CONTINUE AT NEXT INSTRUCTION.
364 000310      001371      DEC      DELCNT      ; TIMEOUT?
365 000314      CLR.STATECHGE      ; NO.
366 000316      042765      000400      000000      BIC      #STATECHGE,CIR(R5)      ; YES.
367 000324      012767      000100      001700      DELAY      100      ; CLEAR INTERRUPT CONDITIONS.
368 000332      104407      000000      65$:      MOV      #100,DELCNT      ; (GIVE SIAS TIME TO CLEAR).
369 000332      104407      000000      BREAK      ; DELAY ( 100 US ).
370 000336      104407      000000      BREAK$,BEGIN      ; BREAK TO MONITOR.
371 000342      005367      001664      BREAK$,BEGIN      ; TEMPORARY RETURN TO MONITOR....
372 000346      001371      DEC      DELCNT      ; THEN CONTINUE AT NEXT INSTRUCTION.
373 000350      042765      000400      000000      BNE      65$      ; TIMEOUT?
374 000354      052765      000004      000002      CLR.STATECHGE      ; NO.
375 000356      052765      000004      000002      BIC      #STATECHGE,CIR(R5)      ; YES.
376 000364      000240      SET.GTS      ; CLEAR STATE CHANGE.
377 000366      000240      BIS      #GTS,SMR(R5)      ; GO TO STANDBY STATE.
378 000370      012767      000100      001634      SCOPE
379 000376      104407      000000      66$:      NOP
380 000376      104407      000000      NOP
381 000376      104407      000000      DELAY      100      ; (GIVE CSBS TIME TO SET).
382 000376      104407      000000      MOV      #100,DELCNT      ; DELAY ( 100 US ).
383 000376      104407      000000      BREAK      ; BREAK TO MONITOR.
384 000376      104407      000000      BREAK$,BEGIN      ; TEMPORARY RETURN TO MONITOR....

```

```

000402 104407 000000' BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
000406 005367 001620 DEC DELCNT ; TIMEOUT?
000412 001371 BNE 66$ ; NO.
; YES.
; TAKE CONTROL SYNCR.
365 000414 SET.CS
000414 052765 000001 000002 BIS #TCS,SMR(R5)
366 000422 WAIT.1 100,ILLMSGE,CIR,T2 ; WAIT ( 100 US ) FOR ILLMSGE = 1.
000422 012757 000100 001602 MOV #100,DELCNT
C00430 0327 5 010000 000000 67$: BIT #ILLMSGE,CIR(R5)
; BIT ILLMSGE = 1?
; YES.
; NO. BREAK TO MONITOR.
000436 001024 BNE T2 ; TEMPORARY RETURN TO MONITOR....
000440 BREAK ;THEN CONTINUE AT NEXT INSTRUCTION.
000440 104407 000000' BREAK$,BEGIN ; TIMEOUT?
000444 104407 000000' BREAK$,BEGIN ; NO.
000450 005367 001556 DEC DELCNT ; YES.
000454 001365 BNE 67$ ; NO ILLEGAL MESSAGE.
; SET UP FOR ERROR CALL.
367 000456 ERROR 46,25 ; SET ERROR CODE TYPE.
000456 004767 001500 JSR PC,ESETUP ; X-REF ERR CODE & DESCRIPTIVE MSG.
000462 012767 000025 177416 MOV #25,ERRTP ; ASCII MESSAGE CALL WITH COMMON HEADER
000470 104403 000000' 002266' MSGN$,BEGIN,MSG46 ; CALL ERROR, PRINT ALL REGS.
000476 HRDR REGTAB
;*****
000476 104405 000000' 002252' HRDR$,BEGIN,REGTAB ;
;*****
000504 000240 NOP
000506 000240 NOP
368
369 000510 T2:
370 000510 042765 177400 000000 BIC #HIBYTE,CIR(R5) ; CLEAR INTERUPT CONDITIONS.
371 000516 SET.CA ; TAKE CONTROL ASYNCR.
000516 052765 000002 000002 BIS #TCA,SMR(R5)
372 000524 DELAY 100 ; (GIVE STATECHGE & CACS TIME TO SET).
000524 012767 000100 001500 MOV #100,DELCNT ; DELAY ( 100 US ).
000532 BREAK ; BREAK TO MONITOR.
000532 104407 000000' BREAK$,BEGIN ; TEMPORARY RETURN TO MONITOR....
000536 104407 000000' BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
000542 005367 001464 DEC DELCNT ; TIMEOUT?
000546 001371 BNE 64$ ; NO.
; YES.
373 000550 SCOPE
000550 000240 NOP
000552 000240 NOP
374 000554 042765 177400 000000 BIC #HIBYTE,CIR(R5) ; CLEAR INTERUPT BITS.
375 000562 012767 000646' 001450 MOV #T3,IRTN ; SET UP INTERUPT RETURN ADDRESS.
376 000570 SET.INTENB ; ENABLE INTERUPT.
000570 052765 000100 000000 BIS #INTENB,CIR(R5)
377 000576 116765 001440 000004 MOV#B LIS1NR,IQR(R5) ; SEND LISTENER ADDRESS.
378 000604 WAIT.INT
000604 010567 177264 MOV R5,SVR5 ; SAVE CONTENTS OF R5.
000610 EXIT ; WAIT FOR INTERUPT - EXIT TO MONITOR.
000610 104400 000000' EXIT$,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
379 000614 ERROR 50,23 ; NO INTERRUPT TO LISTENER ADDRESSING.
000614 004767 001342 JSR PC,ESETUP ; SET UP FOR ERROR CALL.
000620 012767 000023 177260 MOV #23,ERRTP ; SET ERROR CODE TYPE.

```

```

000626 MSGN MSG50 ; X-REF ERR CODE & DESCRIPTIVE MSG.
000626 104403 000000' 002272' MSGN$,BEGIN,MSG50 ;ASCII MESSAGE CALL WITH COMMON HEADLR
000634 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
000634 104405 000000' 002252' HRDR$,BEGIN,REGTAB ;
;*****
000642 000240 NOP
000644 000240 NOP
380
381 000646 T3:
382 000646 T4:
383 000646 SCOPE
000646 000240 NOP
000650 000240 NOP
384 000652 042765 177400 000000 BIC #HIBYTE,CIR(R5) ; CLEAR INTERRUPT BITS.
385 000660 012767 000744' 001352 MOV #T5,IRTN ; SET UP INTERRUPT RETURN ADDRESS.
386 000666 SET,INTENB ; ENABLE INTERRUPT.
000666 052765 000100 000000 BIS #INTENB,CIR(R5)
387 000674 116765 001344 000004 MOV# TALKER,IO(R5) ; ADDRESS TALKER AGAIN.
388 000702 WAIT,INT
000702 010567 177166 MOV R5,SVRS ; SAVE CONTENTS OF R5.
000706 EXIT ; WAIT FOR INTERRUPT - EXIT TO MONITOR.
000706 104400 000000' EXIT$,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.

389 000712 ERROR 53,23 ; NO INTERRUPT TO TALKER ADDRESSING.
000712 004767 001244 JSR PC,ESETUP ; SET UP FOR ERROR CALL.
000716 012767 000023 177162 MOV #23,ERRTYP ; SET ERROR CODE TYPE.
000724 MSGN MSG53 ; X-REF ERR CODE & DESCRIPTIVE MSG.
000724 104403 000000' 002302' MSGN$,BEGIN,MSG53 ;ASCII MESSAGE CALL WITH COMMON HEADER
000732 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
000732 104405 000000' 002252' HRDR$,BEGIN,REGTAB ;
;*****
000740 000240 NOP
000742 000240 NOP
390
391 000744 T5:
392 000744 SET,GTS ; GO TO STANDBY STATE.
000744 052765 000004 000002 BIS #GTS,SMR(R5)
393 000752 DELAY 100 ; (GIVE CSBS TIME TO SET).
000752 012767 000100 001252 MOV #100,DELCNT ; DELAY ( 100 US ).
000760 BREAK ; BREAK TO MONITOR.
000760 104407 000000' BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR...
000764 104407 000000' BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
000770 005367 001236 DEC DELCNT ; TIMEOUT?
000774 001371 BNE 64$ ; NO.
; YES.

394 000776 TST,1 LACS,SMR,T6
000776 032765 010000 000002 BIT #LACS,SMR(R5)
; LACS SET IN SMR ?
001004 001015 BNE T6 ; YES.
; NO.

395 001006 ERROR 52,25 ; NO LACS.
001006 004767 001150 JSR PC,ESETUP ; SET UP FOR ERROR CALL.
001012 012767 000025 177066 MOV #25,ERRTYP ; SET ERROR CODE TYPE.
001020 MSGN MSG52 ; X-REF ERR CODE & DESCRIPTIVE MSG.
001020 104403 000000' 002276' MSGN$,BEGIN,MSG52 ;ASCII MESSAGE CALL WITH COMMON HEADER

```

```

001026                                HRDER  REGTAB                                ; CALL ERROR, PRINT ALL REGS.
;*****
001026 104405 000000' 002252'          HRDER$,BEGIN,REGTAB                                ;
;*****
001034 000240                          NOP
001036 000240                          NOP
396
397 001040                                T6:
398 001040 112765 177777 000004          MOVB   #ALL,IOR(R5)                                ; SEND A DATA BYTE.
399 001046 012767 000100 001156          DELAY  100                                        ; ( GIVE DATAACC TIME TO SET).
001046 012767 000100 001156          MOV    #100,DELCNT                                ; DELAY ( 100 US ).
001054                                64$:
001054 104407 000000'                    BREAK                                     ; BREAK TO MONITGR.
001054 104407 000000'                    BREAK$,BEGIN                                ; TEMPORARY RETURN TO MONITOR....
001060 104407 000000'                    BREAK$,BEGIN                                ; THEN CONTINUE AT NEXT INSTRUCTION.
001064 005367 001142                    DEC    DELCNT                                    ; TIMEOUT?
001070 001371                    BNE   64$                                        ; NO.
; YES.
; CLEAR DATA ACCEPTED.
400 001072                                CLR.DATAACC
001072 042765 100000 000000          BIC   #DATAACC,CIR(R5)
401 001100                                CLR.STATECHGE
001100 042765 000400 000000          BIC   #STATECHGE,CIR(R5)                ; CLEAR STATE CHANGE.
402 001106 105365 000004                    DECB  IOR(R5)                                ; SEND NEXT DATA BYTE.
403 001112                                65$:
001112 012767 000100 001112          DELAY  100                                        ; DELAY ( 100 US ).
001120                                BREAK                                     ; BREAK TO MONITOR.
001120 104407 000000'                    BREAK$,BEGIN                                ; TEMPORARY RETURN TO MONITOR....
001124 104407 000000'                    BREAK$,BEGIN                                ; THEN CONTINUE AT NEXT INSTRUCTION.
001130 005367 001076                    DEC    DELCNT                                    ; TIMEOUT?
001134 001371                    BNE   65$                                        ; NO.
; YES.
404 001136                                TST.0  DATAACC,CIR,T7
001136 032765 100000 000000          BIT   #DATAACC,CIR(R5)
; DATAACC CLEAR IN CIR ?
001144 001415                    BEQ   T7                                        ; YES.
; NO.
405 001146                                ERROR  54,44
001146 004767 001010                    JSR   PC,ESETUP                                ; SET UP FOR ERROR CALL.
001152 012767 000044 176726          MOV   #44,ERRTYP                              ; SET ERROR CODE TYPE.
001160                                MSGN  MSG54
001160 104403 000000' 002306'          MSGN$,BEGIN,MSG54                            ; X-REF ERR CODE & DESCRIPTIVE MSG.
001166                                HRDEP  REGTAB                                ; ASCII MESSAGE CALL WITH COMMON HEADER
; CALL ERROR, PRINT ALL REGS.
;*****
001166 104405 000000' 002252'          HRDER$,BEGIN,REGTAB                                ;
;*****
001174 000240                          NOP
001176 000240                          NOP
406
407 001200                                T7:
408 001200 016567 000004 001042          MOV   IOR(R5),GOT                                ; GET IOR CONTENT.
409 001206 000367 001036                    SWAB  GOT
410 001212 126727 001032 000377          CMPB  GOT,#377                                ; DATA OK?
411 001220 001417                    BEQ   T8                                        ; YES.
412 001222 000367 001022                    SWAB  GOT
413 001226                                ERROR  55,1
001226 004767 000730                    JSR   PC,ESETUP                                ; NO, REBUILD DATA PATTERN READ.
001232 012767 000001 176646          MOV   #1,ERRTYP                              ; DATA CHECK ON IEC TALKER OR LISTENER.
001240                                MSGN  MSG55                                ; SET UP FOR ERROR CALL.
; SET ERROR CODE TYPE.
; X-REF ERR CODE & DESCRIPTIVE MSG.

```

```

001240 104403 000000' 002312' MSGN$,BEGIN,MSG55 ;ASCII MESSAGE CALL WITH COMMON HEADER
001246 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
001246 104405 000000' 002252' HRDR$,BEGIN,REGTAB ;
;*****
001254 000240 NOP
001256 000240 NOP
414
415 001260 T8:
416 001260 TST.1 DATAACC,CIR,T9
001260 032765 100000 000000 BIT #DATAACC,CIR(R5)
; DATAACC SET IN CIR ?
; YES.
; NO.
; NO SECOND "DATA ACCEPTED".
; SET UP FOR ERROR CALL.
; SET ERROR CODE TYPE.
; X-REF ERR CODE & DESCRIPTIVE MSG.
001270 001270 004767 000666 ERROR 56,25
001274 012767 000025 176604 JSR PC,ESETUP
001302 MOV #25,ERRTYP
MSGN MSG56 ; X-REF ERR CODE & DESCRIPTIVE MSG.
001302 104403 000000' 002316' MSGN$,BEGIN,MSG56 ;ASCII MESSAGE CALL WITH COMMON HEADER
001310 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
001310 104405 000000' 002252' HRDR$,BEGIN,REGTAB ;
;*****
001316 000240 NOP
001320 000240 NOP
418
419 001322 T9:
420 001322 016567 000004 000720 MOV IOR(R5),GOT ; GET SECOND DATA BYTE.
421 001330 000367 000714 SWAB GOT ;
422 001334 126727 000710 000376 CMPB GOT,#376 ; IS SECOND DATA BYTE OK?
423 001342 001417 BEQ T10 ; YES.
424 001344 000367 000700 SWAB GOT ; NO, REBUILD PATTERN READ.
425 001350 ERROR 57,1 ; DATA CHECK ON SECOND BYTE.
001350 004767 000606 JSR PC,ESETUP ; SET UP FOR ERROR CALL.
001354 012767 000001 176524 MOV #1,ERRTYP ; SET ERROR CODE TYPE.
001362 MSGN MSG57 ; X-REF ERR CODE & DESCRIPTIVE MSG.
001362 104403 000000' 002322' MSGN$,BEGIN,MSG57 ;ASCII MESSAGE CALL WITH COMMON HEADER
001370 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
001370 104405 000000' 002252' HRDR$,BEGIN,REGTAB ;
;*****
001376 000240 NOP
001400 000240 NOP
426
427
428 ;**
429 ; RANDOM TALKER AND LISTENER DATA TEST.
430 ;
431 ; THIS TEST CAN NOT BE CALLED SEPARATELY, DUE TO THE NECESSARY.
432 ; SETUP TO BRING THE BUS INTO THIS STATE.
433 ;
434 ;**
435 001402 T10:
436 001402 005067 000626 CLR RANMOD ; DISABLE RANDOM MODE.
437 001406 005067 000634 CLR DATBUF ; CLEAR DATA BUFFER.
438 001412 042765 177400 000000 BIC #HIBYTE,CIR(R5) ; CLEAR INTERRUPT BITS.
439 001420 SCOPE

```

START RESTART TEST

```

001420 000240 NOP
001422 000240 NOP
440 001424 012767 001524' 000606 MOV #T12,IRTN ; SET INTERRUPT RETURN ADDRESS.
441 001432 012767 001000 000576 MOV #1000,LOOP ; INIT LOOP COUNTER.
442 001440 T11:
443 001440 042765 177400 000000 BIC #HIBYTE,CIR(R5) ; CLEAR INTERRUPT BITS.
444 001446 SET,INTENB ; ENABLE INTERRUPTS.
001446 052765 000100 000000 BIS #INTENB,CIR(R5)
445 001454 016765 000566 000004 MOV DATBUF,IOR(R5) ; SEND A DATA BYTE.
446 001462 WAIT,INT
001462 010567 176406 MOV R5,SVR5 ; SAVE CONTENTS OF R5.
001466 EXIT ; WAIT FOR INTERRUPT - EXIT TO MONITOR.
001466 104400 000000' EXIT$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

447 001472 ERROR 60,23 ; NO INTERRUPT TO DATA SENDING.
001472 004767 000464 JSR PC,ESETUP ; SET UP FOR ERROR CALL.
001476 012767 000023 176402 MOV #23,ERRTYP ; SET ERROR CODE TYPE.
001504 MSGN MSG60 ; X-REF ERR CODE & DESCRIPTIVE MSG.
001504 104403 000000' 002326' MSGN$,BEGIN,MSG60 ;ASCII MESSAGE CALL WITH COMMON HEADER
001512 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
001512 104405 000000' 002252' HRDR$,BEGIN,REGTAB
;*****
001520 000240 NOP
001522 000240 NOP

448
449 001524 T12:
450 001524 T13:
451 001524 016567 000004 000516 MOV IOR(R5),GOT ; READ BACK DATA BYTE.
452 001532 105067 000512 CLR# GOT ; PREPARE DATA.
453 001536 000367 000506 SWAB GOT ;
454 001542 026767 000502 000476 CMP GOT,DATBUF ; DATA OK?
455 001550 001415 BEQ T14 ; YES.
456 001552 ERROR 62,1 ; DATA CHECK ON IEC BUS.
001552 004767 000404 JSR PC,ESETUP ; SET UP FOR ERROR CALL.
001556 012767 000001 176322 MOV #1,ERRTYP ; SET ERROR CODE TYPE.
001564 MSGN MSG62 ; X-REF ERR CODE & DESCRIPTIVE MSG.
001564 104403 000000' 002332' MSGN$,BEGIN,MSG62 ;ASCII MESSAGE CALL WITH COMMON HEADER
001572 HRDR REGTAB ; CALL ERROR, PRINT ALL REGS.
;*****
001572 104405 000000' 002252' HRDR$,BEGIN,REGTAB
;*****
001600 000240 NOP
001602 000240 NOP

457
458 001604 T14:
459 001604 005767 000424 TST RANMOD ; RANDOM MODE?
460 001610 001011 BNE T15 ; YES.
461 001612 005267 000430 INC DATBUF ; NO. BUILD NEXT DATA PATTERN.
462 001616 026727 000424 000400 CMP DATBUF,#400 ; ALL DATA (0-377) TRIED?
463 001624 001305 BNE T11 ; NO.
464 001626 012767 177777 000400 MOV #-1,RANMOD ; YES. SWITCH TO RANDOM MODE.
465 001634 T15:
466 001634 RAND ; GET A RANDOM NUMBER (PATTERN).
001634 104417 000000' RAND$,BEGIN
467 001640 016767 176210 000400 MOV RANNUM,DATBUF ; MOVE PATTERN ONTO STACK,
468 001646 042767 177400 000372 BIC #177400,DATBUF ; BUILD A BYTE.

```

D2

469	001654	005367	000356		DEC	LOOP		BUMP LOOP COUNT, DONE?
470	001660	001267			BNE	T11		NO.
471	001662	005067	000346		CLR	RANMOD		SWITCH BACK TO NORMAL MODE.
472	001666	012767	000222	000352	MOV	#222,DATBUF		PREPARE LAST DATA PATTERN.
473	001674				SET	LASTBYTE		INDICATE "LAST BYTE".
	001674	052765	000004	000000	BIS	#LASTBYTE,CIR(R5)		
474	001702	016765	000340	000004	MOV	DATBUF,IOR(R5)		LOAD IT.
475	001710				WAIT.1	100,DATAACC,CIR,T16		
	001710	012767	000100	000314	MOV	#100,DELCNT		WAIT ( 100 US ) FOR DATAACC = 1.
	001716	032765	100000	000000	64\$:	BIT	#DATAACC,CIR(R5)	
	001724	001024						BIT DATAACC = 1?
	001726				BNE	T16		YES.
	001726	104407	000000'		BREAK			NO. BREAK TO MONITOR.
	001732	104407	000000'		BREAK\$,BEGIN			TEMPORARY RETURN TO MONITOR....
	001736	005367	000270		BREAK\$,BEGIN			THEN CONTINUE AT NEXT INSTRUCTION.
	001742	001365			DEC	DELCNT		TIMEOUT?
					BNE	64\$		NO.
								YES.
476	001744				ERROR	63,25		NO DATA ACC WHEN SENDING LAST BYTE.
	001744	004767	000212		JSR	PC,ESETUP		SET UP FOR ERROR CALL.
	001750	012767	000025	176130	MOV	#25,ERRTYP		SET ERROR CODE TYPE.
	001756				MSGN	MSG63		X-REF ERR CODE & DESCRIPTIVE MSG.
	001756	104403	000000'	002336'	MSGN\$,BEGIN,MSG63			ASCII MESSAGE CALL WITH COMMON HEADER
	001764				HRDR	REGTAB		CALL ERROR, PRINT ALL REGS.
	001764	104405	000000'	002252'	HRDR\$,BEGIN,REGTAB			*****
	001772	000240			NOP			
	001774	000240			NOP			
477								
478	001776				T16:			
479	001776				TST.1	END,CIR,T17		
	001776	032765	020000	000000	BIT	#END,CIR(R5)		
	002004	001015						END SET IN CIR ?
					BNE	T17		YES.
								NO.
480	002006				ERROR	64,25		NO "END" TO "LAST BYTE".
	002006	004767	000150		JSR	PC,ESETUP		SET UP FOR ERROR CALL.
	002012	012767	000025	176066	MOV	#25,ERRTYP		SET ERROR CODE TYPE.
	002020				MSGN	MSG64		X-REF ERR CODE & DESCRIPTIVE MSG.
	002020	104403	000000'	002342'	MSGN\$,BEGIN,MSG64			ASCII MESSAGE CALL WITH COMMON HEADER
	002026				HRDR	REGTAB		CALL ERROR, PRINT ALL REGS.
	002026	104405	000000'	002252'	HRDR\$,BEGIN,REGTAB			*****
	002034	000240			NOP			
	002036	000240			NOP			
481								
482	002040				T17:			
483	002040	016567	000004	000202	MOV	IOR(R5),GOT		READ BACK DATA BYTE.
484	002046	105067	000176		CLRB	GOT		
485	002052	000367	000172		SWAB	GOT		PREPARE FOR CHECK.
486	002056	026727	000166	000222	CMP	GOT,#222		DATA OK?
487	002064	001415			BEQ	TEND		YES.
488	002066				ERROR	65,1		DATA CHECK ON "LAST BYTE".
	002066	004767	000070		JSR	PC,ESETUP		SET UP FOR ERROR CALL.
	002072	012767	000001	176006	MOV	#1,ERRTYP		SET ERROR CODE TYPE.



E2

```

002100      MSGN      MSG65      ; X-REF ERR CODE & DESCRIPTIVE MSG.
002100 104403 000000' 002346'  MSGN$,BEGIN,MSG65      ;ASCII MESSAGE CALL WITH COMMON HEADER
002106      HRDR      REGTAB      ; CALL ERROR, PRINT ALL REGS.
;*****
002106 104405 000000' 002252'  HRDR$,BEGIN,REGTAB      ;
;*****
002114      NOP
002116      NOP
489
490 002120      TEND;
491 002120      SET,MC      ; MASTER CLEAR.
002120 052765 000040 000000  BIS      #MC,CIR(R5)
492 002126      ENDIT
002126 104413 000000'  ENDIT$,BEGIN      ;SIGNAL END OF ITERATION.
;MONITOR SHALL TEST END OF PASS
493 002132 000167 176066      JMP      RESTRT      ; RESTART NEXT ITERATION.

```

F2

```

496                                     .SBTTL  INTERUPT SERVICE ROUTINE
497
498 002136                               INTSVC: PIRQ   GOIRTN
      002136 000004 000000' 002144'      |-----|
      | PIRQ$,BEGIN,GOIRTN                | QUEUE UP TO CONTINUE AT GOIRTN AND RTI
      |-----|
499
500 002144 016705 175724                 GOIRTN: MOV    SVR5,R5                ; RESTORE R5.
501 002150 142765 177400 000000         BICB   0HIBYTE,CIR(R5)            ; CLEAR INTERUPT BITS.
502 002156 000177 000056                 JMP     0IRTN                    ; RETURN.

```

505  
 506  
 507  
 508  
 509  
 510  
 511 002162 010567 175712  
 512 002166 011567 175710  
 513 002172 010500  
 514 002174 010067 000052  
 515 002200 062700 000002  
 516 002204 010067 000044  
 517 002210 062700 000002  
 518 002214 010067 000036  
 519 002220 062700 000002  
 520 002224 010067 000030  
 521 002230 000207

.SBTTL SUBROUTINES

```

;***
; SUBROUTINE - SET UP FOR ERROR CALL.
;***

```

```

ESETUP: MOV     R5,CSRA           ; SET UP CSR (CIR) ADDRESS.
        MOV     (R5),ACSR       ; SET UP CSR (CIR) CONTENTS.
        MOV     R5,R0           ; SET UP REG POINTERS IN TABLE.
        MOV     R0,CIRPTR
        ADD     #2,R0
        MOV     R0,SMRPTR
        ADD     #2,R0
        MOV     R0,IORPTR
        ADD     #2,R0
        MOV     R0,VSRPTR
        RTS     PC

```

```

524                                     .SBTTL  CONSTANTS AND VARIABLES.
525
526 002232 000000                      DELCNT: 0                ; DELAY COUNTER.
527 002234 000000                      RANMOD: 0                ; RANDOM MODE (0=NOT RANDOM, -1=RANDOM)
528 002236 000000                      LOOP: 0                 ; LOOP COUNTER.
529 002240 000000                      IRTN: 0                 ; INTERRUPT RETURN ADDRESS.
530 002242 000075                      LISTNR: 75              ; ADDRESS OF LISTENER.
531 002244 000135                      TALKER: 135            ; ADDRESS OF TALKER.
532 002246 000000                      DATBUF: 0              ; DATA WORD OR BYTE BUFFER.
533 002250 000000                      GOT: 0                 ; DATA ACTUALLY GOTTEN.
534
535 002252                                REGTAB:                ; REGISTER POINTERS SAVED HERE FOR ERR DUMPING.
536 002252 000000                      CIRPTR: CIR            ; CIR POINTER (AFTER R5 ADDED IN).
537 002254 000002                      SMRPTR: SMR           ; SMR POINTER (AFTER R5 ADDED IN).
538 002256 000004                      IORPTR: IOR           ; IOR POINTER (AFTER R5 ADDED IN).
539 002260 000006                      VSRPTR: VSR           ; VSR POINTER (AFTER R5 ADDED IN).
540 002262 002246                      DBPTR: DATBUF         ; DATBUF (DATA BUFFER) POINTER.
541 002264 177777                      -1                    ; TABLE TERMINATOR.

```

547				.SBTTL	ASCII MESSAGES
548					
549	002266	002352'		MSG46:	EMSG46
550	002270	177777			-1
551	002272	002426'		MSG50:	EMSG50
552	002274	177777			-1
553	002276	002527'		MSG52:	EMSG52
554	002300	177777			-1
555	002302	002630'		MSG53:	EMSG53
556	002304	177777			-1
557	002306	002704'		MSG54:	EMSG54
558	002310	177777			-1
559	002312	003041'		MSG55:	EMSG55
560	002314	177777			-1
561	002316	003136'		MSG56:	EMSG56
562	002320	177777			-1
563	002322	003271'		MSG57:	EMSG57
564	002324	177777			-1
565	002326	003367'		MSG60:	EMSG60
566	002330	177777			-1
567	002332	003464'		MSG62:	EMSG62
568	002334	177777			-1
569	002336	003552'		MSG63:	EMSG63
570	002340	177777			-1
571	002342	003646'		MSG64:	EMSG64
572	002344	177777			-1
573	002346	003710'		MSG65:	EMSG65
574	002350	177777			-1
575					
576	002352	105	122	122	EMSG46: .ASCIZ 'ERROR 46 - "TCS" DOES NOT PRODUCE "ILLMSGE"'
577	002426	105	122	122	EMSG50: .ASCIZ 'ERROR 50 - WHEN SENDING THE LISTENER ADDRESS, NO INTERRUPT OCCURS'
578	002527	105	122	122	EMSG52: .ASCIZ 'ERROR 52 - AFTER LISTENER ADDRESSING AND "GTS", NO "LACS" IS SET'
579	002630	105	122	122	EMSG53: .ASCIZ 'ERROR 53 - NO INTERRUPT TO TALKER ADDRESSING'
580	002704	105	122	122	EMSG54: .ASCII 'ERROR 54 - WHEN SENDING A SECOND DATA BYTE BEFORE READING'
581	002775	045	124	110	.ASCIZ 'AT THE FIRST AGAIN, "DATAACC" CAME UP'
582	003041	105	122	122	EMSG55: .ASCIZ 'ERROR 55 - FIRST DATA BYTE SENT AND RECEIVED IS NOT THE SAME'
583	003136	105	122	122	EMSG56: .ASCII 'ERROR 56 - AFTER READING THE FIRST SENT DATA BYTE, THE SECOND SENT'
584	003240	045	104	111	.ASCIZ 'DID NOT BRING "DATAACC"'
585	003271	105	122	122	EMSG57: .ASCII 'ERROR 57 - SECOND DATA BYTE SENT AND RECEIVED IS NOT THE SAME'
586	003367	105	122	122	EMSG60: .ASCIZ 'ERROR 60 - SENDING OF DATA BYTE DOES NOT PRODUCE AN INTERRUPT'
587	003464	105	122	122	EMSG62: .ASCIZ 'ERROR 62 - DATA BYTES SENT AND RECEIVED ARE NOT EQUAL'
588	003552	105	122	122	EMSG63: .ASCIZ 'ERROR 63 - NO "DATAACC" WHEN SENDING A BYTE WITH "LASTBYTE"'
589	003646	105	122	122	EMSG64: .ASCIZ 'ERROR 64 - NO "END" TO "LASTBYTE"'
590	003710	105	122	122	EMSG65: .ASCIZ 'ERROR 65 - DATA CHECK ON "LASTBYTE"'
591					.EVEN
592					

J2

IEBA DEC/X11 SYSTEM EXERCISER M MACRO M1200 30-MAR-84 11:01 PAGE 23  
END OF PROGRAM

SEQ 22

598			.JB TTL	END OF PROGRAM
599				
600	003754	055036	DDMMY:	23070.
601	003756	001750	HHMM:	1000.
602				
603	003760	000001	PEND:	.END

ACSR	000102R	DATER\$	104404	IOR	000004	POPSP	005726	SR4	000024R
ADDR	000006R	DBPTR	002262R	IORPTR	002256R	POPSP2	022626	START	000224R
ADDR22	001000	DCMMY	003754R	IRTN	002240R	PRHMS\$	000002	STAT	000026R
ALL	177777	DELCNT	002232R	KTPRES	000400	PRTY	000000	STATEC	000400
APTPRE	000200	DVID1	000014R	KTXTND	040000	PRTY0	000000	SVR0	000062R
ASB	000106R	ECCMEM	000100	LACS	010000	PRTY1	000040	SVR1	000064R
ATAT	000104R	EMSG46	002352R	LASTBY	000004	PRTY2	000100	SVR2	000066R
AUTO	000010	EMSG50	002426R	LISINR	002242R	PRTY3	000140	SVR3	000070R
AWAS	000110R	EMSG52	002527R	LOBYTE	000377	PRTY4	000200	SVR4	000072R
BEGIN	000000R	EMSG53	002630R	LOOP	002236R	PRTY5	000240	SVR5	000074R
BIT0	000001	EMSG54	002704R	MAP22\$	104416	PRTY6	000300	SVR6	000076R
BIT1	000002	EMSG55	003041R	MC	000040	PRTY7	000340	SYSCNT	000052R
BIT10	002000	EMSG56	003136R	MODNAM	000000R	PS	177776	TALKER	002244R
BIT11	004000	EMSG57	003271R	MODSP	000224R	PSW	177776	TCA	000002
BIT12	010000	EMSG60	003367R	MSGN\$	104403	PUSH	005746	TCS	000001
BIT13	020000	EMSG62	003464R	MSG\$	104402	PUSH2	024646	TEND	002120R
BIT14	040000	EMSG63	003552R	MSG\$	104401	PWRFLG	000002	TRPDFD	000023
BIT15	100000	EMSG64	003646R	MSG46	002266R	QMON22	000010	TSTIEB	000250R
BIT2	000004	EMSG65	003710R	MSG50	002272R	RAND\$	104417	T1	000250R
BIT3	000010	END	020000	MSG52	002276R	RANMOD	002234R	T10	001402R
BIT4	000020	ENDIT\$	104413	MSG53	002302R	RANNUM	000054R	T11	001440R
BIT5	000040	END\$	104410	MSG54	002306R	REGTAB	002252R	T12	001524R
BIT6	000100	ERRTYP	000100R	MSG55	002312R	RESTR	000224R	T13	001524R
BIT7	000200	ESETUP	002162R	MSG56	002316R	RES1	000056R	T14	001604R
BIT8	000400	EXIT\$	104400	MSG57	002322R	RES2	000060R	T15	001634R
BIT9	001000	GETPA\$	104415	MSG60	002326R	RH70	001000	T16	001776R
BREAK\$	104407	GOIRTN	002144R	MSG62	002332R	RSTRT	000112R	T17	002040R
BR1	000012R	GOT	002250R	MSG63	002336R	R6	000006	T2	000510R
BR2	000013R	GTS	000004	MSG64	002342R	R7	000007	T3	000646R
BTOD\$	104421	GWBUF\$	104414	MSG65	002346R	SACS	000001	T4	000646R
CACS	000400	HMM	003756R	NCPUOP	000020	SBADR	000102R	T5	000744R
CAPRES	000004	LIBYTE	177400	NOPTY	000002	SIAS	020000	T6	001040R
CDATA\$	104412	HRDCNT	000044R	NULL	000000	SIC	000100	T7	001200R
CIR	000000	HRDR\$	104405	OPEN	000000	SMR	000002	T8	001260R
CIRPTR	002252R	HRDPAS	000050R	OTOA\$	104420	SMRPTR	002254R	T9	001322R
CKHNG\$	000001	ICONT	000036R	PARPRE	002000	SOFCNT	000042R	USTACK	000001
CLKPRE	000001	ICOUNT	000040R	PASCNT	000034R	SOFR\$	104406	VECTOR	000010R
CLKSP\$	104422	IDNUM	000122R	PDPF11	000002	SOFPAS	000046R	VSR	000006
CONFIG	000056R	ILLMSG	010000	PDPLSI	020000	SPOINT	000032R	VSRPTR	002260R
CSBS	001000	INDPAR	000040	PDP44	100000	SPSIZ	000040	WASADR	000104R
CSRA	000100R	INIT	000030R	PDP60	004000	SR1	000016R	WDFR	000116R
DATAAC	100000	INTENB	000100	PDP70	010000	SR2	000020R	WDT0	000114R
DATBUF	002246R	INTR	000120R	PEND	003760R	SR3	000022R	XFLAG	000005R
DATCK\$	104411	INTSVC	002136R	PIRQ\$	000004				

. ABS. 000000 000  
003760 001  
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 14148 WORDS ( 56 PAGES)  
DYNAMIC MEMORY: 20060 WORDS ( 77 PAGES)  
ELAPSED TIME: 00:00.35  
XIEBAO,XIEBAO/-SP=DDXCOM.MAC,XIEBAO.MAC

DOCUMENT .....B1  
DOCUMENT .....C1  
DOCUMENT .....D1  
DOCUMENT .....E1  
DOXCOM HEADER .....F1  
DOXCOM HEADER .....G1  
MACRO DEFINITIONS. ....H1  
MACRO DEFINITIONS. ....I1  
EQUATES .....J1  
START - RESTART - TE....K1  
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  
INTERUPT SERVICE ROU....F2  
SUBROUTINES .....G2  
CONSTANTS AND VARIAB....H2  
ASCII MESSAGES .....I2  
END OF PROGRAM .....J2  
SYMBOL TABLE .....K2