

.REM !

IDENTIFICATION

PRODUCT CODE: AC-E914D-MC
PRODUCT NAME: CXIBADO IBV-11A MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

THE IBA IS AN IOMOD THAT EXERCISES THE IBV11-A INSTRUMENTATION BUS INTERFACE (IB BUS). ON START IT DOES A BRIEF LOGIC TEST. ON RESTART AND AFTER ENDPASS, IT EXERCISES THE IBV11-A BY MAKING IT INTERRUPT AND PASS DATA TO ITSELF.

IT IS RECOMMENDED THAT YOU REMOVE THE IB-BUS CABLE FROM THE IBV11 BEFORE YOU RUN THIS MODULE, HOWEVER IF YOU CHOCSE NOT TO AND YOU DO GET ERRORS, THE ERRORS COULD BE CAUSED FROM SOME DEVICE ON THE IB-BUS AND NOT THE IBV11-A.

2.0 REQUIREMENTS

HARDWARE: ONE IBV11-A.
STORAGE:: IBA REQUIRES:
1. DECIMAL WORDS: 417
2. OCTAL WORDS: 0641
3. OCTAL BYTES: 1502

3.0 PASS DEFINITION

ONE PASS OF THE IBA MODULE CONSISTS OF 17700 (OCTAL) DATA TRANSFERS TO AND FROM THE IBV11-A.

4.0 EXECUTION TIME

ONE PASS OF THE IBA MODULE RUNNING ALONE TAKES APPROXIMATELY ONE MINUTE.

5.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:
DEVADR:160150, VECTOR:640, BR1:4, DEVCNT:1, SR1:0

REQUIRED PARAMETERS:
NONE.

6.0 DEVICE/OUTPUT SET-UP:

NONE.

7.0 MODULE OPERATION

TEST	SEQUENCE
1. (START)	BIT EXERCISE CSR, DATA REGISTER.
2. (RESTART)	DATA TRANSFERS
A.	SET INTERRUPT ENABLE AND TCS.
E.	IBV INTERRUPTS TO "TKRSRV"
C.	SET "TON" AND "LON", CLEAR TCS
D.	IBV INTERRUPTS TO "TKRSRV"
E.	DATA PATTERN GENERATED AND LOADED INTO IBD (DATA REGISTER)
F.	DATA GETS "GATED" ONTO IB-BUS AND RECEIVED BY IBD THUS GATING A "LNR" INTERRUPT.
G.	"LNR" INTERRUPTS TO "LNRSRV"
H.	DATA RECEIVED IS COMPARED AGAINST DATA SENT, IF UNEQUAL, AN ERROR IS REPORTED.
I.	IF ALL DATA TRANSFERS HAVE BEEN MADE, AN "ENDPASS" IS REPORTED, ELSE PROGRAM CONTROL IS RETURNED TO STEP D.

8.0 OPERATION OPTIONS

VALID SRI VALUES

SRI BIT	ENABLE/DISABLE	FUNCTION
0	0	ENABLE RANDOM PATTERN TRANSFERS
0	1	ENABLE COMPLEMENT PATTERN TRANSFERS. PATTERN IN LOCATION "PATTERN" IS COMPLEMENTED EACH TIME IT IS USED.

9.0 NON-STANDARD PRINTOUTS

ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE
DEC/X11 DOCUMENT.

!

```
126 .TITLE IBAD DEC/X11 SYSTEM EXERCISER MODULE
127 ; DDXCOM VERSION 6 23-MAY-78
128 .LIST RIN
129 ;*****
130 REGIN: 000000 041111 042101 040
131 MODNAM: .ASCII /IBAD / ;MODULE NAME.
132 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
133 ADDR: 160150*0 ;1ST DEVICE ADDR.
134 VECOR: 640*0 ;1ST DEVICE VECTOR.
135 BR1: .BYTE PRTV6+0 ;1ST BR LEVEL.
136 BR2: .BYTE PRTV6+0 ;2ND BR LEVEL.
137 DIVD1: +1 ;DEVICE INDICATOR 1.
138 SR1: OPEN ;SWITCH REGISTER 1.
139 SR2: OPEN ;SWITCH REGISTER 2.
140 SR3: OPEN ;SWITCH REGISTER 3.
141 SR4: OPEN ;SWITCH REGISTER 4.
142 ;*****
143 STAT: 140000 ;STATUS WORD.
144 INIT: START ;MODULE START ADDR.
145 SPOINT: MODSP ;MODULE STACK POINTER.
146 PASCNT: 0 ;PASS COUNTER.
147 ICNT: 177000 ;# OF ITERATIONS PER PASS=177000
148 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
149 SDFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
150 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
151 SDFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
152 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
153 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
154 RANDUM: 0 ;HOLDS RANDOM # WHEN RAND MACRG IS CALLED
155 CONFIG: 0 ;RESERVED FOR MONITOR USE
156 RES1: 0 ;RESERVED FOR MONITOR USE
157 RES2: 0 ;RESERVED FOR MONITOR USE
158 SVR0: OPEN ;LOC TO SAVE R0.
159 SVR1: OPEN ;LOC TO SAVE R1.
160 SVR2: OPEN ;LOC TO SAVE R2.
161 SVR3: OPEN ;LOC TO SAVE R3.
162 SVR4: OPEN ;LOC TO SAVE R4.
163 SVR5: OPEN ;LOC TO SAVE R5.
164 SVR6: OPEN ;LOC TO SAVE R6.
165 CSADR: OPEN ;ADDR OF CURRENT CSR.
166 ACSR: OPEN ;ADDR OF GOOD DATA, OR
167 WASADR: OPEN ;CONTENTS OF CSR.
168 ASADR: OPEN ;ADDR OF BAD DATA, OR
169 ASSTAT: OPEN ;STATUS REG CONTENTS.
170 ERRVTP: 0 ;TYPE OF ERROR.
171 ASB: 0 ;EXPECTED DATA.
172 ANAS: OPEN ;ACTUAL DATA.
173 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
174 MDIR: OPEN ;WORDS TO MEMORY PER ITERATION
175 MDIR: OPEN ;WORDS FROM MEMORY PER ITERATION
176 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
177 IDNUM: 141 ;MODULE IDENTIFICATION NUMBER=141
178 MODSP: 0
179 ;*****
```

```
180 ;MODULE REQUIRED REGISTERS - SET UP BY THIS MODULE.
181
182 IBS: .WORD 160150 ;CONTROL AND STATUS
183 IBD: .WORD 160152 ;DATA REGISTER
184
185 VECTA: .WORD 640 ;ERROR INTR. VECTOR
186 PRA: .WORD 642 ;SRQ INTR. VECTOR
187 VECTB: .WORD 644 ;TKR OR CMD INTR. VECTOR
188 PRB: .WORD 646 ;LNR INTR. VECTOR
189 VECTC: .WORD 650
190 PRC: .WORD 652
191 VECTD: .WORD 654
192 PRD: .WORD 656
193
194 ;CONSTANTS AND STORAGE LOCATIONS
195
196 CNT: .WORD 0 ;ICNT-1
197 PATERN: .WORD 252 ;CURRENT PATTERN BEING
198 ;TRANSFERRED BETWEEN TKR AND LNR
199 ;ON IBV11 BUS.
200
201 RANA: 123456 ;RANDOM NUMBERS GENERATED
202 RANB: 071234 ;BY RANDOM NUMBER GENERATOR
203 ;TO GET "PATERN" IF NOT
204 ;INHIBITED BY "SRI"
205
206
207 START:
208 MOV ICNT,CNT ;SAVE ITERATION COUNT
209 DEC CNT ;DEFINE NEXT TO LAST ITERATION
210 MOV ADDR,IBS ;GET BASE CSR ADDR.
211 MOV IBS,IBD ;NOW FORM DBR ADDR.
212 ADD #2,IBD ;CSR+2.
213
214 MOV VECTOR,R0 ;NOW LETS FIX VECTOR ADDRESS
215 R0,VECTA ;/SET THIS VECTOR ADDR.
216 ADD #2,R0 ;/UPDATE ADDR FOR NEXT VECTOR.
217 R0,PRA ;/SET THIS VECTOR ADDR.
218 ADD #2,R0 ;/UPDATE ADDR FOR NEXT VECTOR.
219 R0,VECTB ;/SET THIS VECTOR ADDR.
220 ADD #2,R0 ;/UPDATE ADDR FOR NEXT VECTOR.
221 R0,PRB ;/SET THIS VECTOR ADDR.
222 ADD #2,R0 ;/UPDATE ADDR FOR NEXT VECTOR.
223 R0,VECTC ;/SET THIS VECTOR ADDR.
224 ADD #2,R0 ;/UPDATE ADDR FOR NEXT VECTOR.
225 R0,PRC ;/SET THIS VECTOR ADDR.
226 ADD #2,R0 ;/UPDATE ADDR FOR NEXT VECTOR.
227 R0,VECTD ;/SET THIS VECTOR ADDR.
228 ADD #2,R0 ;/UPDATE ADDR FOR NEXT VECTOR.
229 R0,PRD ;/SET THIS VECTOR ADDR.
230
231 .REM !
232
233
234
235 IBS - IBV11-A CONTROL & STATUS REGISTER
```

236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291

IBS INSTRUMENT BUS CONTROL & STATUS REGISTER 16X150
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SRQ ER2 ER1 EPO 0 CMD TKR LNR ACC IE TON LON IBC PEM EOP TCS
R R R R R R R R W W W W W W W W

R MEANS THIS BIT MAY ONLY BE READ BY THE LSI-11.
W MEANS THIS BIT MAY BE BOTH READ AND WRITTEN BY THE LSI-11.

IBD - IBV11-A DATA REGISTER

THE HIGH BYTE MONITORS THE 5 I/O CONTROL LINES AND THE 3 I/O
DATA HANDSHAKE LINES. THE DATA ON THESE LINES IS LATCHED TO
REMAIN STABLE WHILE BEING READ. THE LOW BYTE MONITORS THE I/O
DIO LINES DIRECTLY WHEN READ. DATA WRITTEN INTO THE LOW BYTE IS
LATCHED INTO A BUFFER AND SENT OUT ON THE I/O DIO LINES IF TON
OR TCS AND ATN ARE SET. THE BUFFER IS ALWAYS CLEARED BY DAC.

IBD INSTRUMENT BUS DATA REGISTER 16X152
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
EOI ATN IFC REN SRQ RFD DAV DAC IOB IO7 IO6 IO5 IO4 IO3 IO2 IO1
R R R R R R R R W W W W W W W W

R MEANS THIS BIT MAY ONLY BE READ BY THE LSI-11.
W MEANS THIS BIT MAY BE BOTH READ AND WRITTEN BY THE LSI-11.

!
;* LOGIC TEST #1 BE SURE AN IBV11-A EXISTS AT THE
;* SPECIFIED ADDR. IF NOT IBV11-A, THEN A DEC/X11
;* SYS ERROR WILL OCCUR.

```
000414* 005777 177604 LOG1: TST @IBS ;ADDRESS THE IBV. IF SYS ERROR  
;OCCURS, THEN IBV DID NOT RESPOND  
;WHEN ADDRESSED.  
;MAKE SURE THAT THE ADDRESS  
;FORMED BY THE SWITCH PACK ON THE  
;OPTION AGREE WITH THE ADDRESS  
;THIS SOFTWARE HAS IN "ADDR".
```

;* LOGIC TEST #2
IN THIS TEST WE ARE GOING TO SET
TCS (CSR BIT0), EOP (CSR BIT01),
REM (BIT02), LON (CSR BIT04),
TON (BIT05), AND ACC (CSR BIT07)
NOW, TCS SHOULD CAUSE CMD (CSR BIT10)
AND DAC (IBD BIT8) TO SET, EOP
SHOULD CAUSE EOI (IBD BIT15) TO SET.
REM SHOULD CAUSE REN (IBD BIT12) TO SET.

292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347

;* LON SHOULD CAUSE LNR (CSR BIT8) TO SET.
;* TON SHOULD CAUSE TKR (CSR BIT9) TO SET.
;* TCS SHOULD ALSO CAUSE ATN (IBD BIT14) TO SET.
;* WE ARE GOING TO CHECK THAT ALL THESE BITS SET.
;* IF YOU GET ANYTHING LIKE AN ERROR1 OR ERROR2 OR SRQ
;* SET - READ DESCRIPTION PRECEDING "ERRSRV".

```
000420* 012777 000267 177576 LOG2: MOV #267,@IBS ;LOAD IBS WITH ACC, TON, LON, EOP  
;AND TCS  
MOV @IBS,ACSR ;READ BACK IBS  
MOV #002267,ASTAT ;EXPECT ACC, TON, LON, EOP, TCS, REM  
;AND CMD TO  
;BE SET  
CMP ACSR,ASTAT ;DID THEY ALL SET?  
REQ IS ;YES - THEN NEXT CHECK.  
MOV @IBS,CSRA ;NO - RECORD CSR'S ADDR.  
MOV #25,ERRTYP  
*****  
HDRS$,BEGIN,NULL ;CSR PROBLEM WITH CSR BITS ASTAT=GOOD, ACSR=BAD  
*****  
000474* 017767 177526 177400 1S: MOV @IBD,ACSR ;NOW READ DATA REGISTER  
MOV #152000,ASTAT ;EXPECT EOI, ATN, REN AND  
;RFD TO BE SET  
CMP ACSR,ASTAT ;DID THEY ALL SET?  
REQ S ;YES - NEXT TEST.  
MOV @IBD,CSRA ;NO - RECORD DBR'S ADDR.  
MOV #25,ERRTYP  
*****  
HDRS$,BEGIN,NULL ;DBR PROBLEM WITH DBR BITS ASTAT=GOOD, ACSR=BAD  
*****  
000542* 005077 177456 2S: CLR @IBS ;CLEAR THE CSR.
```

;* LOGIC TEST #3
IN THIS TEST WE'LL MAKE
SURE THE INTERRUPT ENABLE (BIT6)
WILL SET AND CLEAR.

```
000546* 104407 000000* LOG3: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....  
000552* 104407 000000* BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
000556* 012777 000100 177440 MOV #BIT6,@IBS ;SET INTERRUPT ENABLE.  
000564* 032777 000100 177432 BIT #BIT6,@IBS ;DID IT SET?  
000572* 001017 ;YES - LETS TRY CLEARING IT.  
000574* 017767 177424 177300 MOV @IBS,ACSR ;RECORD CSR FOR ERROR TYPEOUT.  
000602* 012767 000100 177274 MOV #BIT6,ASTAT ;RECORD SHOULD BE.  
000610* 016767 177410 177262 MOV @IBS,CSRA ;RECORD IBV ADDR.  
000616* 012767 000027 177262 MOV #27,ERRTYP ;IE FAILED TO CLEAR
```

```
348 ;*****  
349 HRDRS,BEGIN,NULL ;INTERRUPT ENABLE FAILED TO CLEAR  
350 ;*****  
351 000624* 104405 000000* 000000  
352 CLR @IBS ;CLEAR INTERRUPT ENABLE.  
353 MOV @IBS,ACSR ;READ CSR - ALL SHOULD BE CLEAR.  
354 BEQ 2$ ;IF CLEAR - NEXT TEST.  
355 CLR A$TAT ;EXPECT CLEAR CSR.  
356 MOV IBS,CSRA ;RECORD IBV ADDR.  
357  
358 MOV #27,ERRTYP  
359 ;*****  
360 HRDRS,BEGIN,NULL ;IF OR CSR NOT CLEAR  
361 ;*****  
362 000666* 104405 000000* 000000  
363 ;*****  
364 000674* 2$:  
365 ;*  
366 ;* LOGIC TEST #4 IN THIS TEST WE'LL CHECK IBC.  
367 ;* FIRST WE'LL LOAD THE CSR  
368 ;* WITH EOP, REM, LON, TON, AND ACC.  
369 ;* NEXT WE'LL DO (SET) ON IBC  
370 ;* (CSR BIT03). THIS SHOULD CLEAR  
371 ;* EOP, REM, LON, TON, AND ACC, THEN SET  
372 ;* TCS, AND CMD IN CSR.  
373 ;*  
374 ;* THE ONLY TRICKY THING HERE IS  
375 ;* WATCHING THAT IBC TIMES OUT  
376 ;* IN APPROXIMATELY 125 US.  
377 ;*  
378 ;*  
379 000674* LOG4:  
380 000674* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...  
381 000700* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
382  
383 000704* 012777 000266 177312 MOV #266,@IBS ;LOAD IBS WITH ACC,TON,LOW  
384 ;AND TCS  
385 000712* 052777 000010 177304 BIS #BIT3,@IBS ;NOW DO AN IBC.  
386 000720* 012701 000062 MOV #50,R1 ;SET TO TIME OUT IF IBC DOESN'T  
387 000724* 032777 000010 177272 1$: BIT #BIT3,@IBS ;IBC TIMEOUT YET?  
388 BEQ 2$ ;YES - CONTINUE  
389 000734* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...  
390 000740* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.  
391 000744* 005301 DEC R1 ;OUR LITTLE COUNTER TIME OUT YET?  
392 000746* 001366 BNE 1$ ;NO - CONTINUE  
393 ;TIME LOOP EXCEED 125 US - IBC SHOULD HAVE CLEARED.  
394 000750* 017767 177250 177124 MOV @IBS,ACSR ;RECORD CONTENTS OF CSR.  
395 000756* 012767 002001 177120 MOV #2001,ASTAT ;SHOULD READ.  
396 000764* 016767 177234 177106 MOV IBS,CSRA ;RECORD ADDR. OF IBV.  
397  
398 MOV #25,ERRTYP  
399 ;*****  
400 001000* 104405 000000* 000000 HRDRS,BEGIN,NULL ;IBC (BIT3) FAILED TO TIME OUT  
401 ;*****  
402 001006* 017767 177212 177066 2$: MOV @IBS,ACSR ;IBC TIMED-OUT - NOW READ CSR.  
403 ;*****
```

```
404 001014* 012767 002001 177062 MOV #2001,ASTAT ;THE ONLY BITS THAT SHOULD BE  
405 ;SET IN THE CSR ARE CMD (BIT10)  
406 001022* 026767 177054 177054 CMP ACSR,ASTAT ;AND TCS (BIT0)  
407 001030* 001411 BEQ 3$ ;DID IT WORK OUT OK?  
408 ;YES - NEXT TEST.  
409  
410 001032* 016767 177166 177040 MOV IBS,CSRA ;RECORD IBV ADDR.  
411  
412 001040* 012767 000025 177040 MOV #25,ERRTYP  
413 ;*****  
414 001046* 104405 000000* 000000 HRDRS,BEGIN,NULL ;CSR WORKS AFTER ON BC, ACSR=FA$T, A$TAT=SHOULD BE  
415 ;*****  
416 001054* 005077 177144 3$: CLR @IBS ;CLEAR THE CSR.  
417 ;ENTER HERE ON RESTART  
418  
419 001060* RE$TRT:  
420 ;BEFORE DECLARING ENDPASS.  
421 CLR @IBS ;CLEAR THE CSR.  
422 001064* 005077 177140 MOV #TKRSRV,@VECTC ;SET UP TKR * CMD INTR.  
423 001072* 012777 000200 177142 MOV #200,@PRC ;PRIORITY ON INTERRUPT.  
424 001076* 012777 001222 177136 MOV #LNRSRV,@VECTD ;SETUP LNR INTR.  
425 001106* 012777 000200 177132 MOV #200,@PRD ;PRIORITY ON INTERRUPT  
426 001114* 012777 001380 177106 MOV #ERRSRV,@VECTA ;POINT TO ERPRD SERVICE ROUTINE  
427 001122* 012777 000200 177102 MOV #200,@PRA ;PRIORITY ON INTERRUPT.  
428 001130* 012777 001350 177076 MOV #ERRSRV,@VECTB ;POINT TO ERROR ROUTINE ON SRQ  
429 001136* 012777 000200 177072 MOV #200,@PRB ;PRIORITY ON INTERRUPT  
430  
431 001144* 012777 000101 177052 ERRRLP: MOV #BIT6@BIT0,@IBS ;SET INTERRUPT ENABLE AND TCS -  
432 ;THIS WILL CAUSE IBV TO INTERRUPT  
433 ;AFTER IT HAS CONTROL OF THE IB-BUS.  
434 001152* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.  
435 ;*  
436 ;* TKRSRV - THIS INTERRUPT SERVICE ROUTINE WILL TAKE  
437 ;* CARE OF "CMD" AND "TKR" INTERRUPTS FROM THE  
438 ;* IBV11. WHEN TEST IS STARTED, WE ISSUED A  
439 ;* "TCS" SO THE FIRST THING THAT SHOULD HAPPEN IS A  
440 ;* "CMD" INTR. WHEN WE GET THAT, WE'LL SET "TON"  
441 ;* AND "LON" TO TRANSFER DATA.  
442 ;* WHEN TON GETS SET WE'LL GET INTERRUPTED  
443 ;* RIGHT BACK HERE WHERE WE'LL XMITT DATA. WHEN  
444 ;* DATA IS XMITTED, WE'LL GET A LNR INTERRUPT TO  
445 ;* "LNRSRV" WHERE WE'LL DELAY READING DATA. "ACC"  
446 ;* IN THE IBS PREVENTS US FROM BEING FLOODED WITH "TKR"  
447 ;* INTR. - THAT IS WE MUST ACTUALLY READ THE DATA FROM  
448 ;* THE IBD BEFORE A "TKR" INTR. HITS US AGAIN.  
449 ;*  
450 ;*  
451 ;*  
452 001156* 032777 002000 177040 TKRSRV: BIT #BIT10,@IBS ;IS THIS A "CMD" INTERRUPT?  
453 001164* 001404 000360 177030 BEQ 1$ ;NO - "TKR" SERVICE  
454 001166* 012777 000360 177030 MOV #BIT7@BIT6@BIT5@BIT4,@IBS ;YES - CLEAR TCS (CLEARS ATN) SET FOR  
455 ;TKR AND LNR (WITH ACC) INTERRUPTS.  
456 ;THIS INTERRUPT DID NEED IMMEDIATE  
457 ;ATTENTION, ALL OTHERS CAN BE DELAYED.  
458 001174* 000002 RTI ;EXIT - TKR INTERRUPT NEXT.  
459 ;*****
```

```

460 001176* 1S:
461
462 001176* 000004 000000* 001204* PIRQS,BEGIN,2S ; QUEUE UP TO CONTINUE AT 2S AND RTI
463
464
465 001204* 004767 000212 177010 2S: JSR PC,PATGEN ;GO GET A PATTERN TO SEND
466 001210* 016777 177036* MOV PATTERN,@IBD ;LOAD PATTERN.
467 001216* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
468
469
470 ;LNRSRV THIS ROUTINE RECEIVES "LNR" INTERRUPTS FROM THE IBV11
471 ; IT WILL DELAY, COMPARE DATA XMITTED TO DATA RECEIVED,
472 ; IF BAD, WE'LL REPORT AN ERROR. IF GOOD
473 ; IT ALLOWS THE NEXT "TKR" INTERRUPT.
474 ; WE'LL ALSO CHECK FOR "END PASS".
475
476 001222* 042777 000100 176774 LNRSRV: BIC #100,@IBS ;TMP CLEAR OF INTERRUPT ENABLR.
477
478 001230* 000004 000000* 001236* PIRQS,BEGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTI
479
480 001236* 1S:
481 001236* 016767 177010 176640 MOV PATTERN,ASTAT ;RECORD CURRENT PATTERN.
482 001244* 017767 176756 176630 MOV @IBD,ACSR ;READ DATA FROM DBR.
483 001252* 042767 177400 176622 BIC #177400,ACSR ;STRIP UNUSED DATA BITS
484 001260* 026767 176620 176614 CMP ASTAT,ACSR ;DATA SENT = DATA RECEIVED?
485 001266* 001411 BEQ 2S ;YES - CONTINUE
486 001270* 016767 176732 176602 MOV IBD,CSRA ;RECORD ADDR. OF DATA REG.
487
488 001276* 012767 000001 176602 MOV #1,ERRTYP
489 *****
490 001304* 104405 000000* 000000 HRDRS,BEGIN,NULL ;BAD DATA TRANSFER-ASTAT=DATA SENT ;ACSR=DATA RECEIVED
491 *****
492
493 001312* 026767 176732 176520 2S: CMP CNT,ICOUNT ;DONE ENOUGH XFERS?
494 001320* 001407 BEQ 3S ;YES - REPORT END PASS
495 001322* 104413 000000* ENDIRS,BEGIN ;SIGNAL END OF ITERATION.
496 ;MONITOR SHALL TEST END OF PASS
497 001326* 052777 000100 176670 BIS #BIT6,@IBS ;NO - RESET INTERRUPT ENABLR - THIS
498 ;WILL CAUSE A "TKR" INTERRUPT
499 ;FOR NEXT DATA.
500 001334* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
501
502 001340* 005077 176660 3S: CLR @IBS ;CLEAR THE CSR.
503 001344* 104413 000000* ENDIRS,BEGIN ;SIGNAL END OF ITERATION.
504 ;MONITOR SHALL TEST END OF PASS
505
506
507
508
509
510
511
512
513
514
515
ERRSRV THIS ROUTINE TAKES CARE OF ERROR INTERRUPTS
AS WELL AS INTERRUPTS THAT ARE FATAL
TO THIS PIECE OF SOFTWARE. IT WILL
RECORD THE CSR, CLEAR THE CSR - DELAY (VIA
A PIRQ) THEN REPORT THE ERROR.
SINCE THIS TYPE OF ERROR SHOULD NEVER
OCCUR, JUST IN CASE YOU GET ONE, WE
WON'T DROP THE MODULE. IN THIS WAY, YOU

```

```

516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
CAN ACCUMULATE DATA ON THE PROBLEM.
WE WILL HOWEVER, RESET THE TRANSFER
SEQUENCE FROM THE POINT OF SETTING "TCS"
TO GET A "CMD" INTERRUPT.
TYPES OF ERRORS THAT GOT US HERE AND THEIR
CAUSES:
1. SRQ INTERRUPT (BIT15 SET IN IBS)
THIS POINTS TO SOME DEVICE AT THE END OF THE
IB BUS. DISCONNECT THE CABLE FROM
THE IBV11. IF YOU STILL GET A SRQ INTERRUPT,
YOU'VE GOT A PROBLEM.
2. ER1 INTERRUPT (BIT13 SET IN IBS)
THIS BIT IS FATAL TO THE IBV11 WHEN SET.
THE IB BUS IS SCREWED-UP.
THERE ARE TWO CAUSES FOR IT:
A. ATN, IFC OR REN ARE ASSERTED ON THE IB BUS
BY SOMEONE OTHER THAN THIS IBV11. TRY
DISCONNECTING THE IB BUS CABLE FROM THE
IBV11 (THE CUSTOMER MAY HAVE SOME #SIRD
OPTION ON THE IB BUS THAT IS IN DIRECT
CONFLICT WITH THE IBV11 SPEC.). IF THIS
DOESN'T HELP, START LOOKING FOR A PROBLEM
IN THE GATES THAT GENERATE AN ER1.
B. ATN, IFC OR REN ARE ASSERTED ON THE IB BUS
BY THE IBV11, BUT ARE NOT RECEIVED BACK
BY IT. I'VE ALREADY CHECKED THIS CONDITION
ONCE IN THE LOGIC TESTS - BUT IT WAS A
ONE-TIME CHECK AND MAYBE EXERCISING THIS
BABY A LOT CAUSED A PROBLEM IN THAT CIRCUITRY.
NOW TO TRY AND TELL THE DIFFERENCE
BETWEEN A AND B:
IF TCS IS SET IN THE CSR AND ATN IS NOT
SET - B IS THE PROBLEM.
IF TCS IS CLEAR BUT ATN IS SET STEP A IS
THE PROBLEM.
NOW I DON'T USE REN OR IFC IN THE DATA
TRANSFERS SO STEP A IS MOST LIKELY TO
BE THE PROBLEM - EXAMINE THE CSR TO SEE IF
EITHER OR BOTH ARE SET.
3. ER2 INTERRUPT (BIT14 SET)
ERROR 2'S ARE CAUSED BY A TALKER ON
THE BUS AND NO LISTENERS.
I DON'T LET THAT HAPPEN AS I MAKE THE
IBV11 A LISTENER AS WELL AS A TALKER.
YOU BETTER LOOK BACK AT THE CIRCUITRY THAT
GENERATES AN ERROR 2 - ITS PAD!

```


XIBADO.P11 12-OCT-78 12:00 CROSS REFERENCE TABLE -- USER SYMBOLS

LHRSRV	001222R	425	476#																
LOG1	000474R	275#																	
LOG2	000470R	300#																	
LOG3	000546R	335																	
LOG4	000674R	379																	
MAP22S	= 104416	180#																	
MODNAM	= 000000R	131#																	
MODDSP	= 000224R	145	178#																
MSGNS	= 104403	180#																	
MSGSS	= 104402	180#																	
MSGSS	= 104401	180#																	
NULL	= 000000	180#	312	324	349	360	400	414	490	583	162	163	164	165					
OPEN	= 000000	132	138	139	140	141	158	159	160	161									
		167	169	171	172	174	175	176	180#										
OTDAS	= 104420	180#																	
PASCNT	= 000034R	146																	
PATERN	= 000252R	199#	466	481	599*	606*	608*												
PATGRN	= 001422R	465	597#																
PRGOS	= 000006R	180#	462	478	576														
PPSP	= 005726	180#																	
PPSP2	= 022626	180#																	
PR	= 000242R	187	218*	428*															
PRB	= 000236R	189	222*	430*															
PRC	= 000242R	191	226*	424*															
PRD	= 000246R	193	230*	426*															
PRTY	= 000000	180#																	
PRTY0	= 000000	180#																	
PRTY1	= 000040	180#																	
PRTY2	= 000100	180#																	
PRTY3	= 000140	180#																	
PRTY4	= 000200	180#																	
PRTY5	= 000240	180#																	
PRTY6	= 000300	135	136	180#															
PRTY7	= 000340	180#																	
PS	= 177346	180#																	
PSW	= 177776	180#																	
PUSH	= 005746	180#																	
PUSH2	= 004129	180#																	
RANA	= 000148R	203	602	603*	604*	606													
RANB	= 000256R	204	602*	604	605*														
RANDS	= 104417	180#																	
RANMUM	= 000054R	173																	
RESHT	= 001060R	173	420#																
RES1	= 000056R	156#																	
RES2	= 000060R	157#																	
RES3	= 000112R	173																	
SADR	= 000112R	166																	
SDFCNT	= 000042R	149																	
SDFERS	= 104406	180#																	
SDFPIS	= 000046R	151																	
SDFINT	= 000018R	145																	
SPSIZ	= 000040	1	178																
SR1	= 000016R	138	597																
SR2	= 000020R	140																	
SR3	= 000014R	140																	
SR4	= 000024R	141																	

XIBADO.P11 12-OCT-78 12:00 CROSS REFERENCE TABLE -- USER SYMBOLS

START	000260R	144	208#																
STAT	= 000026R	143																	
SVR0	= 000062R	158																	
SVR1	= 000064R	159																	
SVR2	= 000066R	160																	
SVR3	= 000070R	161																	
SVR4	= 000072R	162																	
SVR5	= 000074R	163																	
SVR6	= 000076R	164																	
SYSCNT	= 000052R	151																	
TKRSRV	= 001156R	423	453#																
TRPDFD	= 000022R	180#																	
VECTA	= 000230R	186	216*	427*															
VECTB	= 000234R	189	220*	429*															
VECTC	= 000240R	190	224*	423*															
VECTD	= 000244R	192	228*	425*															
VECTOR	= 000010R	134	215																
WASADR	= 000104R	128																	
WDFR	= 000116R	175																	
WDT0	= 000114R	174																	
XFLAG	= 000005R	132#																	

. AFS. 000000 000
001502 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0
XIBADO, XIBADO/SOL/CRF:SYM=DDXCOM, XIBADO
RUN-TIME: 1 2 .2 SECONDS
RUN-TIME RATIO: 15/4=3.7
CORE USED: 7K (13 PAGES)