

IDENTIFICATION

PRODUCT CODE: AC-F884R-MC
PRODUCT NAME: CXBEAR0 M7855 BUS TESTER 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 CONSIDERED 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 MAY 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) 1975,1978 DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
3.0	PASS DEFINITION
4.0	EXECUTION TIME
5.0	CONFIGURATION REQUIREMENTS
6.0	DEVICE/OPTION SETUP
7.0	TEST SEQUENCE
8.0	OPERATION OPTIONS
9.0	NON STANDARD PRINTOUTS

1.0 ABSTRACT

BEA IS AN IOMOD THAT CAN HANDLE FROM 1 TO 12 UNIBUS EXERCISERS. THE MODULE WILL HAVE THE UBF(S) DOING DATA(S), DATA(S), DATA(S), AND DATA(S). THEN CHECKS FOR CORRECT DATA TRANSFERS. THESE TRANSFERS ARE DONE FIRST ON AN NPR LEVEL AND INTERRUPTS WITH A BR7, THEN THE REQUEST IS SEQUENTIALLY LOWERED TO A BR4.

2.0 REQUIREMENTS

HARDWARE:

1 TO 12 UBF(S); WITH MORE THAN ONE UBF, ALL SHOULD HAVE W1 JUMPER EXCEPT THE UBF THAT IS THE FURTHEST FROM THE PROCESSOR ELECTRICALLY.

SOFTWARE:

STORAGE:: BEA REQUIRES:

- 1. DECIMAL WORDS: 1371
- 2. OCTAL WORDS: 02533
- 3. OCTAL BYTES: 5266

3.0 PASS DEFINITION

1. ITERATION CONSISTS OF SETTING UP ONE OR MORE UBF(S) TO DO DATA TRANSFERS. THEN CHECKING THOSE TRANSFERS. 1 PASS WILL EQUAL 12,000 ITERATIONS.

4.0 EXECUTION TIME

ONE PASS WILL RUN IN APPROX. 1 MINUTE, RUNNING ALONE ON AN 11/45.

5.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS: DEVADR=170000; VECTOR=510; BR1=7, BR2=6; DEVCNT=1
REQUIRED PARAMETERS: NONE

6.0 DEVICE/OPTION SETUP

1 TO 12 UBF(S); WITH MORE THAN ONE UBF, ALL SHOULD HAVE W1 JUMPER EXCEPT THE UBF THAT IS THE FURTHEST FROM THE PROCESSOR ELECTRICALLY.

7.0 TEST SEQUENCE

- A. LOAD DEVICE INTR VECTORS AND GET READ AND WRITE BUFFER SIZES.
- B. INITIALIZE REGISTERS.
- C. GET PHYSICAL ADDRESSES FOR WRITE AND READ BUFFERS.
- D. CLEAR DEVICE REGISTERS AND WRITE BUFFER AREA; AND SET UP READ BUFFER AREA.
- E. CHECK FOR DEVICES AND IF NONE LEFT, GO TO H.
- F. LOAD DEVICE REGISTERS FOR DESIRED TRANSFER.
- G. GO BACK TO E.
- H. SET OFF ALL DEVICES SIMULTANEOUSLY.
- I. WAIT FOR THEIR INTERRUPTS.
- J. IF NOT ALL DEVICES INTERRUPTED, GO BACK TO I.
- K. CHECK THE BUFFER AREAS FOR CORRECT TRANSFERS.
- L. ROTATE REQUEST LEVELS AND DATA PATTERNS.
- M. IF THIS IS NOT THE FINAL ITERATION, GO BACK TO D.
- N. INDICATE END OF PASS AND GO BACK TO B.

8.0 OPERATION OPTIONS

THE FIRST REGISTER ADDRESS OF EACH USE PROCEEDS IN INCREMENTS OF 20. EVERY BIT OF DVID1 CORRESPONDS TO A PARTICULAR USE ADDRESS BEING PRESENT; I.E. BIT 0 SET = USE ADDRESS OF 170000, BIT 1 SET = USE ADDRESS OF 170020, BIT 2 SET = USE ADDRESS OF 170040, ETC.

THE USE(S) CAN BE ON THE BUS IN ANY ADDRESSING SEQUENCE. THE ONLY REQUIREMENT IS THAT DVID1 REFLECTS THE ADDRESSES OF EXISTING USE(S).

9.0 NON-STANDARD PRINTOUTS

ALL PRINTOUT HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT WITH THE FOLLOWING ADDITIONS PRINTED BELOW IT:

1. FOR DATA TRANSFER ERRORS (DATA/P OF DATA/B) THE CONTENTS OF THE FOLLOWING REGISTERS PRE-PRINTED AS WELL AS THE MEMORY LOCATION AND CONTENTS. THERE SHOULD BE A DISCREPANCY BETWEEN REGR AND CONTENTS OF MEMORY ON ERRORS.
BEDB BFCC RERA BECR1 BECR2 MEM-ADDR MEM-CONTENTS
2. FOR "INTERRUPTS ON ERROR - NOT ON DONE" THE ABOVE REGISTERS WILL ALSO BE PRINTED OUT EXCEPT DISCARD THE MEMORY ADDRESS AND CONTENTS SINCE IT IS NOT PREVIOUSLY SET UP.
3. FOR "NOT ALL DEVICES INTERRUPTING" ERROR, ONLY TWO ADDITIONAL REGISTERS ARE PRINTED - DV AND MASK. DV IS A TEMPORARY STORAGE LOCATION FOR DVID1 AND MASK IS THE LOCATION OF INDICATING WHICH DEVICES INTERRUPTED. BIT COMPARISON OF THESE TWO REGISTERS WILL INDICATE WHICH DEVICES DID OR DID NOT INTERRUPT.

DV MASK

└

```

203 000000-
204 000000-
205
206
207
208
209 000000-
210 000000- 042502 041101 040
211 000005- 000
212 000006- 170000
213 000010- 000510
214 000012- 340
215 000013- 300
216 000014- 000001
217 000016- 000000
218 000020- 000000
219 000022- 000000
220 000024- 000000
221
222 000026- 140000
223 000030- 000410-
224 000032- 000224-
225 000034- 000000
226 000036- 012000
227 000040- 000000
228 000042- 000000
229 000044- 000000
230 000046- 000000
231 000050- 000000
232 000052- 000000
233 000054- 000000
234 000056-
235 000058- 000000
236 000060- 000000
237 000062- 000000
238 000064- 000000
239 000066- 000000
240 000070- 000000
241 000072- 000000
242 000074- 000000
243 000076- 000000
244 000100- 000000
245 000102-
246 000104- 000000
247
248 000104- 000000
249 000106-
250 000108- 000000
251 000110- 000000
252 000112- 000534-
253 000114- 000000
254 000116- 000000
255 000120- 000000
256 000122- 000073
257
258

```

```

IOMOD <CREAB >,170000,510,7,6,12000,73
MODULE 140000,BEAB,170000,510,7,6,12000,73
;
;TITLE BEAR DEC/X11 SYSTEM EXERCISER MODULE
;DDICOM VERSION 6 23-MAV-78
;LIST BIN
;*****
BEGIN:
MODNAM: -ASCII /BEAB / ;MODULE NAME.
XFLAG: -BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR: 170000+0 ;1ST DEVICE ADDR.
VECTOR: 510+0 ;1ST DEVIC VECTOR.
BR1: -BYTE PRTV7+0 ;1ST RR LEVEL-
; -BYTE PRTV6+0 ;2ND RR LEVEL-
DVID1: +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: 140000 ;STATUS WORD.
INIT: START ;MODULE START ADDR.
SP0INT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER
ICOUNT: 12000 ;# OF ITERATIONS PER PASS=12000
SOFCNT: 0 ;LOC TO COUNT ITERATIONS
HRPCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
SOPPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
HRPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
RANNUM: 0 ;# OF SYS ERRORS ACCUMULATED
CONFIG: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;RESERVED FOR MONITOR USE
SVR0: OPEN ;RESERVED FOR MONITOR USE
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.
ACSPR: OPEN ;ADDR OF GOOD DATA, OR
;CONTENTS OF CSR.
WASADR: OPEN ;ADDR OF BAD DATA, OR
ASTAT: OPEN ;STATUS REG CONTENTS.
ERRTYP: ;TYPE OF ERROR
ASR: OPEN ;EXPECTED DATA.
AAS: OPEN ;ACTUAL DATA.
RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
WDPR: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: 73 ;MODULE IDENTIFICATION NUMBER=73
;*****
-NLIST SPSIZ ;MODULE STACK STARTS HERE.

```

```

259
260
261
262 000224-
263
264

```

```

;WORD 0
;LIST
;ENDR
MODSP:
;*****

```

```

265
266
267
268 000224 000000
269 000226 137777
270 000230 000000
271 000233 100000
272 000234 000000
273 000236 170014
274 000240 000000
275 000242 000000
276 000244 000000
277 000246 170000
278 000250 000740
279 000252 000360
280 000254 000240
281 000256 000170
282 000260 000140
283 000262 000120
284 000264 000000
285 000266 000000
286 000270 000000
287 000272 000000
288 000274 000000
289 000276 000000
290 000300 000000
291 000302 000000
292 000304 000000
293 000306 000000
294 000310 000000
295 000312 000000
296 000314 000000
297 000316 000352
298 000320 000000
299 000322 000000
300 000324 004324
301 000326 000000
302 000330 000000
303 000332 000360
304 000334 000000
305 000336 000000
306 000338 000000
307 000342 000000
308 000344 004100
309 000346 000000
310 000350 022500
311 000352 003100
312 000354 000060
313 000356 007740
314 000360 000000
315 000362 000000
316 000364 000000
317 000366 000000
318 000368 000000
319 000370 000000
320 000372 000374

```

```

.ENABL AMA
P1: 0 ;1ST DATA PATTERN
P2: 137777 ;2ND DATA PATTERN
P3: 0 ;3RD DATA PATTERN
P4: 100000 ;4TH DATA PATTERN
ROTCNT: 0 ;COUNTER FOR # OF TIMES DATA ROTATED
STMLGN: 170014 ;ADDR TO SET OFF ALL DEVS SIMULTANEOUSLY
DV: 0 ;TEMP STORAGE FOR DVID1
MASK: 0 ;USED TO DROP ANY DEV NOT INTERRUPTING
MORE: 0 ;WORKING STORAGE FOR DVID1
B1215: 170000 ;USED TO CLEAR BITS 12 THRU 15
BUFSZ: 740 ;SIZE FOR 1 DEV
360 ;SIZE FOR 2 DEVS
240 ;SIZE FOR 3 DEVS
170 ;SIZE FOR 4 DEVS
140 ;SIZE FOR 5 DEVS
120 ;SIZE FOR 6 DEVS
DEVcnt: 0 ;TOTAL NUMBER OF DEVS
DODI: 0 ;DATA OR DATI INDICATOR
SVDDDI: 0 ;STORAGE FOR DODI
SAVRI: 0 ;LOC TO SAVE R1
SAVR5: 0 ;LOC TO SAVE R5
BTCC: 0 ;CYCLE COUNT FOR A BYTE TRANSFER
WCC: 0 ;CYCLE COUNT FOR A DATA TRANSFER
RCC: 0 ;CYCLE COUNT FOR A DATI TRANSFER
DOCNT: 0 ;# OF DEVS DOING A DATO/B(S)
DTCNT: 0 ;# OF DEVS DOING A DATI/P(S)
EABIT: 0 ;USED TO SET EXTENDED MEMORY ADDR BITS
RBUF: 0 ;SIZE OF BUFFER FOR DATI/P(S)
WRBUF: 0 ;SIZE OF BUFFER FOR DATO/B(S)
WRUFVA: 0 ;VIRTUAL ADDR OF WRITE BUFFER
WRUFPA: 0 ;PHYSICAL ADDR OF WRITE BUFFER
WRUFEA: 0 ;EXT MEM ADDR BITS SET BY MONITOR
RRUFVA: 0 ;VIRTUAL ADDR OF READ BUFFER
RRUFPA: 0 ;PHYSICAL ADDR OF READ BUFFER
RRUFEA: 0 ;EXT MEM ADDR BITS SET BY MONITOR
RBUFSZ: 360 ;TOTAL SIZE OF BUFFER RESERVED FOR READS
RRAFDR: 0 ;USED TO POINT TO LOC WITHIN RBUFSZ
ENDRBF: 0 ;USED FOR LAST ADDR(PHYSICAL) OF READ BUFF
WRRAFDR: 0 ;USED TO POINT TO LOC WITHIN WRBUFSZ
ENDWRBF: 0 ;USED FOR LAST ADDR(PHYSICAL) OF WRITE RUFF
XPR1: 4100 ;FUN 2-DATI-INTR
XPR2: 2500 ;FUN 2-DATO-INTR
XPR3: 22500 ;FUN 1-DATI/NO
XPR4: 3100 ;FUN 1-DATO-INTR
ROVLVL: 60 ;USED TO SET REQUEST LEVELS OF DEVS
ERRBIT: 7740 ;USED TO CHECK IF ANY ERROR BITS SET
DR: 0 ;DEV DATA REG CONTENTS
CC: 0 ;DEV CYCLE COUNT CONTENTS
DOR: 0 ;DEV ADDR REG CONTENTS
CR1: 0 ;DEV CR1 REG CONTENTS
CR2: 0 ;DEV CR2 REG CONTENTS
BADMEM: BADR ;BAD MEMORY ADDR

```

```

321 000374 000000
322 000376 177777
323 000400 000000
324 000400 000240
325 000404 000774
326 000404 000774
327 000406 000000
328
329 000410 012737 000361 000114
330 000416 012737 000361 000116
331 000424 012737 000001 000120
332 000432 013700 000014
333 000440 006200
334 000440 001412
335 000442 103375
336 000444 062737 000361 000114
337 000452 062737 000361 000116
338 000460 005237 000120
339 000464 000764
340 000466 013737 000014 000240
341 000474 043737 000246 000240
342 000502 013737 000240 000242
343 000510 013737 000240 000244
344 000516 001002
345 000520 000137 001404
346 000524
347 000524 004737 001546
348 000530 004737 001410
349
350
351
352
353
354
355
356
357
358
359
360
361 000534
362 000534 104415 000000 000324
363 000542 013737 000326 000336
364 000550 062737 000736 000336
365
366 000556 104415 000000 000316
367 000564 013737 000320 000342
368 000572 062737 000736 000342
369
370
371 000600 004737 002234
372
373 000604 012737 000001 000266
374
375 000612 005037 000234
376 000616 012737 000224 000272

```

```

BADR: 0 ;CONTENTS OF BAD MEMORY
DVMASK: 177777 ;THIS POINTER IS USED SO THE NEXT
PTR1: DV ;TWO LOCATIONS WILL BE PRINTED OUT FOR
PTR2: MASK ;AN ERR WHICH DEVICES DID NOT INTERRUPT
TOCNT: 1 ;TIME OUT COUNTER
START: MOV #361,WDTO ;AT LEAST 361 WORDS TO MEM/ITERATION
MOV #361,WDFR ;AT LEAST 361 WORDS FROM MEM/ITERATION
MOV DVID1,R0 ;AT LEAST INTERRUPT /ITERATION
2S: ASR R0 ;SHIFT IN A COUNT
JCC 3S ;RR OUT IF NONE LEFT
ADD #361,WDTO ;GO BACK IF NO BIT IN THIS POSITION
ADD #361,WDFR ;361 MORE WORDS TO MEM
LNC INTR ;361 MORE WORDS FROM MEM
BR 2S ;MORE INTERRUPT
MOV DVID1,DV ;GO CHECK FOR MORE
BIC B1215,DV ;MOVE DVID1 TO TEMP STORAGE
MOV DV,MASK ;ENSURE BITS 12-15 ARE CLEARED
MOV DV,MORE ;INIT MASK BITS EQUAL TO DVID1
BNE IS ;THEN SFT UP WORKING AREA
JMP DROP ;GO TO IS IF THERE ARE DEVS
ELSE DROP THE MODULE
1S: JSR PC,CLRREG ;CLEAR ALL DEV REGS
JSR PC,SETUP ;SET UP AVAILABLE REG LOCATIONS AND
;GET BUFFER SIZES FOR DATO(S) & DATI(S)
;*****
;NOTE: IN THE FOLLOWING COMMENTS THE TERM READ IS USED
;TO DENOTE DATI OR DATI AND THE TERM WRITE DENOTES
;*****
;THIS IS THE LOCATION THE PROGRAM WILL PROCEED
;WITH AFTER AN END OF PASS
NESTRT: GETPAS,REGIN,RRUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RRUFVA
MOV RRUFPA,ENDRBF ;MOVE READ BUFF ADDR TO END-OF-READ BUFF REG
ADD #736,ENDRBF ;ADD 736 TO GET LAST ADDR OF BUFFER
GETPAS,REGIN,WRUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT WRUFVA
MOV WRUFPA,ENDWRBF ;MOVE WRITE BUFF ADDR TO END-OF-WRITE BUFF REG
ADD #736,ENDWRBF ;ADD 736 TO GET LAST ADDR OF BUFFER
JSR PC,FARITS ;SET BITS 0 & 1 OF PEA EQUAL TO
;BITS 4 & 5 OF RUFVA RESPECTIVELY
MOV #1,DODI ;INIT XPR INDICATOR TO 1 SO THE
;1ST INC WILL SIGNIFY A DATI
CLR ROTCNT ;CLEAR PATTERN ROTATION COUNT
MOV #P1,SAVRI ;INIT SAVRI TO 1ST PATTERN

```

```

377 000624 012737 000344 000274 MOV #XPR1,SAVR5 ;INIT SAVRS TO 1ST XFER FUNCTION
378 000632 012737 000060 000354 MOV #60,RQLVL ;INIT REQUEST LEVEL TO NPP/INTR BPT
379
380 ;THIS IS THE LOCATION THE PROGRAM WILL PROCEED WITH
381 ;AFTER AN ITERATION
382
383 000640 REREST:
384 000640 023737 000242 000240 CMP MASK,DV ;# OF DEVS THAT SHOULD INTR=# OF DEVS THAT DID?
385 000646 001402 BRQ IS ;IF EQUAL,GO TO 2S
386 000650 004737 JSR PC,SETUP ;ELSE GRT NEW DEV COUNTS & BUFFER SIZES
387
388 000654 013737 000266 000270 1S: MOV DDDI,SVDODI ;SAVE DDDI SETTING
389 000662 013737 000242 000244 MOV MASK,MORE ;RESET MORE BITS
390 000670 013737 000242 000240 MOV MASK,DV ;IF ANY DEVS DROPPED, ALTER DV
391 000676 001002 BNE ZS ;IF NOT, CONTINUE ON
392 000700 000137 001404 JMP DROP ;ELSE DROP MODULE
393
394 000704 004737 002402 2S: JSR PC,RESDAT ;RESTORE DATA IN READ BUFFER
395
396 000710 013700 000006 MOV ADDR,RO ;AND CLEAR WRITE BUFFER
397 000714 013737 000320 000340 MOV WRBFFA,WRBFFADR ;PUT DEVICE ADR INTO REG. 0
398 000730 163737 000314 000340 MOV WRBP,WRBFFADR ;USE WRBFFADR AS WRITE BUFF ADDR
399 000736 163737 000312 000334 MOV RRBFFA,RRBFFADR ;INIT TO WRITE BUFF LESS WRBP
400 000736 163737 000312 000334 SUB RRBFF,RRBFFADR ;USE RRBFFADR AS READ BUFF ADDR
401 ;INIT TO READ BUFF LESS RBPFF
402
403 000744 3S: ASR MORE ;CHECK FOR ANOTHER DEV
404 000750 103404 BCS 4S ;IF THERE, CONTINUE
405 000752 001440 GO ;AND IF NOT, GO TO GO
406 000760 062700 000020 ADD #20,RO ;ELSE ADD 20 FOR NEXT ADDR
407 000766 000771 BR 4S ;AND CHECK FOR MORE
408 000762 004737 001742 JSR PC,CHEKPX ;SEE IF R1 & R5 NEED INITIALIZATION
409 000766 005237 000266 INC DDDI ;INC XFER INDICATOR
410 000770 032737 000001 BIT #R10,DDDI ;IF BIT 0 OF DDDI IS 1
411 001000 001003 BNE ZS ;GO SET UP REGS FOR DATO(S)
412 001002 004737 JSR PC,RRRGS ;ELSE SET UP REGS FOR DATI(S)
413 001006 000402 BR 5S ;GO LOAD THE REGISTERS
414
415 001010 004737 002016 5S: JSR PC,WRREGS ;SET REGS FOR WRITE XFER INFO
416
417 ;AT THIS POINT THE DEVICE REGISTERS WILL BE LOADED TO DO
418 ;TRANSFERS
419
420 6S:
421 001014 MOV (R1)+(R0)+ ;SET UP DATA BUFFER REG
422 001014 MOV (R2)+(R0)+ ;SET UP CYCLE COUNT REG.
423 001020 MOV (R3)+(R0)+ ;SET UP ADDR REG
424 001022 MOV CABIT,10(R0) ;SET UP EXT ADDR BITS IN CR2
425 001030 MOV (R5)+(R0) ;SET UP CR1
426 001036 MOV RQLVL,(R0) ;SET REQUEST LEVELS
427 001036 ADD #12,RO ;ADD 12 TO GET NEXT DEV ADDR
428
429 001042 MOV R1,SAVR1 ;SAVE CONTENTS OF R1(LAST DATA PATTERN)
430 001046 MOV R5,SAVR5 ;SAVE CONTENTS OF R5(LAST XFER INSTR.)
431 001052 BR 3S ;GO LOAD ANOTHER DEVICE

```

```

433 001054 GO:
434 001054 005037 000406 CLR TDCNT ;CLEAR TIME OUT COUNT
435 001060 005037 000242 CLR MASK ;CLEAR ALL BITS IN MASK
436 ;TO BE RESET IN INTR SERV RTNS
437 001064 005277 177146 INC @SIMLGO ;SET OFF ALL DEVS
438
439 001070 BRK:
440 001074 104407 000000 BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR----
441 001100 104407 000000 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
442 001106 001411 000240 000242 CMTV DR,MASK ;ALL DEVS INTR?
443 001110 005337 000406 BNE DATACK ;IF YES, GO TO DATACK
444 001114 001365 DEC TDCNT ;ELSE DEC TIME OUT COUNT
445 001124 104407 000000 003122 BNE BRK ;AND BREAK IF NOT 0
446 001124 005004 MSGNS,BPGIN,NOINTR ;ASCII MESSAGE CALL WITH COMMON HEADED
447 001126 004737 JSR PC,GETDEV ;REPORT NON-INTR'G DEVICES
448
449 001132 DATACK:
450 001132 013737 000240 000244 MOV DV,MORE ;SET UP MORE
451 001140 013737 000270 000266 MOV SVDODI,DDDI ;RESTORE DDDI TO SETTING WHEN
452 ;LOADING 1ST DEV
453 001146 013700 000006 MOV ADDR,RO ;INIT RO TO 1ST DEV ADDR
454 001152 012737 004324 000334 MOV #RFDIN,RRBFFADR ;RBFADR=1ST LOC IN READ BUFFER
455 001160 012737 003362 000340 MOV #WRTOUT,WRBFFADR ;WRBFFADR=1ST LOC IN WRITE BUFFER
456 001166 005003 CLR R3 ;USED IN DODATA RTN
457
458 001170 MORDEV:
459 001174 103405 ASR MORE ;CHECK FOR OTHER DEVS
460 001176 001001 BCS 2S ;IF THERE CONTINUE
461 001200 000444 BNE 1S ;IF SOME LEFT,GO CHECK FOR MORE
462 001202 BR NPASS ;ELSE GO TO END OF PASS RTN
463
464 1S:
465 001206 062700 000020 ADD #20,RO ;GET NEXT DEV ADDR
466 001206 000770 BR MORDEV ;GO CHECK FOR THAT DEV
467
468 2S:
469 001210 105760 TSTR 6(R0) ;IS READY BIT SET IN CR1?
470 001214 100407 BMI 3S ;IF SET, CONTINUE
471 001216 104403 MSGNS,BPGIN,NOSET ;ASCII MESSAGE CALL WITH COMMON HEADER
472 001222 004737 CLR R4 ;R4 INVALID IN ERR REPORT
473 001226 004737 JSR PC,RDDATA ;ELSE TYPE OUT CONTENTS OF ALL DEV REGS
474 001232 000426 BR NXT ;AND GO CHECK FOR NEXT DEV
475
476 3S:
477 001234 033760 BIT EPRITS,16(R0) ;SEE IF ANY ERROR BITS SET
478 001242 001407 BNE 4S ;IF NONE SET,CONTINUE
479 001244 104403 MSGNS,BPGIN,ERRSET ;ASCII MESSAGE CALL WITH COMMON HEADER
480 001252 005004 CLR R4 ;R4 INVALID IN ERR REPORT
481 001254 004737 JSR PC,RDDATA ;ELSE TYPE OUT CONTENTS OF ALL DEV REGS
482 001260 000413 BR 4S ;AND GO CHECK FOR NEXT DEV
483
484 4S:
485 001262 005237 INC DDDI ;INC XFER INDICATOR
486 001264 000001 BIT #R10,DDDI ;PART OR DATO?
487 001270 001003 BNE ZS ;IF DATO,GO DO IT
488 001276 004737 JSR PC,DIDATA ;CHECK DATI XFRS
489 001302 000402 BR NXT ;CHECK FOR NEXT DEV
490
491 5S:
492 001304 004737 JSR PC,DODATA ;CHECK DATO XFRS
493 001310 002162 BR 5S ;CHECK FOR NEXT DEV ADDR
494
495 NXT:
496 001310 062700 000020 ADD #20,RO

```



```
489 001314 000137 001170 JMP MORDEV ;GO CHECK FOR ANOTHER DEV
490 001320 001404 NPASS: JSR PC,CLRREG ;CLEAR ALL DEV RECS
491 001320 004737 JSR PC,ROTREG ;SHIFT REQUEST TO NEXT RR LEVEL
492 001324 004737 JSR PC,ROTREG ;HAS LAST XFER BEEN DONE?
493 001330 022737 CMP #XFR4,SAVR5 ;IF NOT, CONTINUE
494 001336 001010 BNE IS ;*****PC*****
;*****PC*****
;AFTER P2 & P4 HAVE BEEN ROTATED 16 TIMES, ALLOW THE POINTER
;R1 TO INCREMENT THRU THE END OF THE READ BUFFER BUT DON'T ALLOW ROTATION
;*****PC*****
501 001340 022737 000020 000234 CMP #16,ROTCNT ;ELSE HAS P2&P4 BEEN ROTATED 16 TIMES?
502 001346 001404 BEQ IS ;IF NOT, GO TO 55
503 001350 005237 000234 INC ROTCNT ;IF YES DON'T ROTATE PATTERNS ANYMORE
504 001354 004737 002342 JSR PC,ROTPTAT ;ELSE INC ROTATION COUNT, AND
505 001360 032737 000001 000264 1S: ROTATE DATA PATTERNS
506 001366 001402 BIT #R10,DEV CNT ;IS THERE AN ODD # OF DEVS?
507 001370 004737 JSR PC,SWAPCC ;IF NOT, GO TO 55
508 001374 104413 000000 2S: ;WITH WBUF FOR CORRECT XFER INFO
509 001374 000137 000640 ENDS,REGIN ;SIGNAL END OF ITERATION
510 001400 000137 000640 JMP REREST ;MONITOR SHALL TEST END OF PASS
511 001404 104410 000000 DRUP: ;ELSE GO TO REREST
512 001404 104410 000000 ENDS,REGIN ;*****PC*****
;*****PC*****
;THIS ROUTINE WILL SET UP THE DEVICE INTO VECTORS, COUNT THE
;NUMBER OF DEVICES, SET THE TRANSFER SIZE OF THE READ AND WRITE
;BUFFERS AND SET UP THE READ AND WRITE CYCLE COUNTS.
;*****PC*****
529 001410 005037 000306 SETUP: CLR DICNT ;CLEAR REG COUNTING DEVS DOING DATI(S)
530 001414 005037 000304 CLR DOCNT ;CLEAR REG COUNTING DEVS DOING DATO(S)
531 001420 012700 002742 MOV #ISPI,RO ;SET RO TO 1ST INTR SERVICE RTN
532 001424 012700 000010 MOV VECOR,R1 ;SET R1 TO 1ST INTR VEC ADDR
533 001430 012737 000001 000266 MOV #1,DDDI ;INIT XFR INDICATOR TO 1
534 001436 006237 000244 CKMORE: ASR MORE ;CHECK FOR A DEVICE
535 001440 103406 BFC IS ;IF THERE, CONTINUE
536 001444 001420 BFC IS ;IF NONE LEFT AT ALL, GET OUT
537 001446 062700 000010 ADD #10,RO ;ELSE INC INTR SERV POINTER BY 10
538 001450 062700 000004 ADD #4,R1 ;AND INC INTR VEC LOC BY 4
539 001456 000767 RR CKMORE ;GO BACK AND CHECK FOR ANOTHER DEV
540 001460 005237 000266 1S: INC DDDI ;INC XFER INDICATOR
541 001464 004737 JSR PC,INCCNT ;INC DATI OR DATO COUNT
542 001470 010021 MOV RO,(R1)+ ;MOVE ADDR OF INTR SERV RTN TO VEC LOC
```

```
545 001472 113721 000012 MOV#R1,(R1)+ ;MOVE R1 VALUE TO PSW VEC
546 001476 105721 TSTR (R1)+ ;GET BACK TO AN EVEN ADDR
547 001500 062700 000010 ADD #10,RO ;ADD 10 TO RO FOR NEXT INTR SERV RTN
548 001504 000754 RR CKMORE ;SEE IF THERE ARE MORE DEVS
549 001506 013737 000266 2S: MOV DDDI,DEV CNT ;DEV XFER INDICATOR HAS # OF DEVS
550 001514 005337 000264 DFC DEV CNT ;+ 1, SO DEC DEV CNT BY 1
551 001520 012700 000246 GETRUF: MOV #BUFSZ-2,RO ;SET RO TO SIZE TABLE
552 001524 005001 CLR R1 ;CLEAR R1
553 001526 005720 1S: TST (RO)+ ;MOVE POINTER TO NEXT VALUE
554 001530 005701 R1 INC COUNTER ;INC COUNTER
555 001534 020137 CMP #1,DICNT ;# OF DATI DEVS ?
556 001536 103773 BLO IS ;IF NOT, INC COUNTER & POINTER
557 001540 011037 MOV (RO),RBUF ;ELSE RO HAS SIZE OF READ BUFF PER DEV
558 001544 011037 MOV (RO),RCC ;1 WORD = 2 LOCATIONS (OR BYTES)
559 001550 006237 ASR RCC ;AND ALSO IS DATI XFER COUNT
560 001554 005437 NEG RCC ;WHEN IT IS HALVED
561 001560 005737 TST DOCNT ;GET 2'S COMP FOR DEV CC REG
562 001564 001405 BEQ 2S ;IF DOCNT IS 0
563 001566 020137 CMP #1,DOCNT ;GO USE THE VALUE RO POINTS TO
564 001572 001402 BFC R1 ;R1=# OF DATO DEVS ?
565 001574 005301 BFC R1 ;YES, CONTINUE
566 001576 005740 TST -(RO) ;NO, DEC R1
567 001600 011037 2S: MOV (RO),WBUF ;RO HAS SIZE OF WRITE BUFF PER DEV
568 001604 011037 MOV (RO),WCC ;AND ALSO IS DATO XFER COUNT
569 001610 006237 ASR WCC ;WHEN IT IS HALVED
570 001614 005437 NEG WCC ;GET 2'S COMP FOR DEV CC REG
571 001620 000207 RTS PC ;RETURN
;*****PC*****
;*****PC*****
;THIS ROUTINE WILL COUNT THE NUMBER OF DEVICES DOING DATI(S)
;AND THOSE DOING DATO(S)
;*****PC*****
587 001622 032737 000001 000266 INCCNT: BIT #R10,DDDI ;IF BIT 0 IS 1
588 001630 001003 BNE INCDG ;GO INC DOCNT
589 001632 005237 000306 INC DICNT ;ELSE INC DICNT
590 001636 000402 BR EXINC ;AND EXIT RTN
591 001640 005237 000304 INCDG: INC DOCNT ;INC DOCNT
592 001644 000207 RTS PC ;RETURN
;*****PC*****
;*****PC*****
```

```
601 ;THIS ROUTINE WILL CLEAR ALL THE REGISTERS OF ALL EXISTING DEVICES
602 ;AS DEFINED BY DVID1
603 ;*****
604 ;*****
605
606 001646- CLRRREG:
607 001646- 013737 000240- 000244- MOV DV, MORE ;RESTORE MORE
608 001654- 013701 000006- MOV ADDR, R1 ;INIT R1 TO 1ST DEV ADDR
609
610 001660- 1S:
611 001660- 006237 000244- ASR MDRF ;IS THERE A DEVICE?
612 001664- 103404 BCS 32 ;IF VCS GO SET IT UP
613 001670- 062701 000020 ADD #20, R1 ;IF MORE LEFT - GET OUT
614 001674- 000771 BR IS ;ELSE INC R1 TO NEXT DEV ADDR
615 ;AND GO SEE IF IT'S THERE
616
617 001700- 005011 000002 2S:
618 001704- 005061 000004 CLR (R1) ;CLEAR DATA BUFFER REG
619 001710- 005061 000006 CLR 2(R1) ;CLEAR CYCLE COUNT REG
620 001714- 005061 000010 CLR 4(R1) ;CLEAR BUFFER ADDR REG
621 001720- 005061 000016 CLR 6(R1) ;CLEAR CR1 REG
622 001724- 062701 000020 CLR 10(R1) ;CLEAR ERROR CLEAR REG
623 001730- 000753 ADD #20, R1 ;CLEAR CR2 REG
624 001732- 013737 000240- 000244- BR IS ;INC R1 TO NEXT DEV ADDR
625 ;GO CHECK FOR ANOTHER DEV
626
627 001740- 000207 RTS PC ;RESTORE MORE
628 ;RETURN
629
630 ;*****
631 ;*****
632 ;THIS ROUTINE WILL KEEP TRACK OF THE DATA PATTERNS AND TRANSFER
633 ;FUNCTIONS USED TO LOAD THE DEVICES
634 ;*****
635 ;*****
636
637 001742- CHECKPX:
638 001742- 013701 000272- MOV SAVR1, R1 ;RESTORE R1 TO PATTERN IN SAVR1
639 001746- 013705 000274- MOV SAVR5, R5 ;RESTORE R5 TO XFR FUNCTION IN SAVR5
640 001750- 027705 000352- CMP #R4, R5 ;DOES R5 POINT TO LAST XFR?
641 001756- 100002 BPL IS ;IF NOT, CONTINUE
642 001760- 012705 000344- MOV #XFR1, R5 ;ELSE INIT TO 1ST XFR
643
644 001764- 1S:
645 001770- 022701 000232- CMP #P4, R1 ;IF R1 DOES NOT EXCEED LAST PATTERN
646 001774- 100011 BPL 3S ;ADDRESS EXIT ROUTINE
647 001778- 022737 000020 000234- CMP #16, ROTCNT ;HAVE P2&P4 BEEN ROTATED 16 TIMES?
648 002000- 001003 BNE #R4, R5 ;IF NOT, INIT R1
649 002006- 100002 CBR #R4, R1 ;DOES R1 LAST ADDR OF READ BUFF?
650 002010- 022010 BPL 3S ;IF LESS, GO TO EXINIT
651
652 002010- 012701 000224- 2S: MOV #P1, R1 ;ELSE INIT R1 TO 1ST DATA PATTERN
653
654 002014- 000207 RTS PC ;RETURN
655 ;*****
656 ;*****
```

```
657 ;*****
658 ;THIS ROUTINE WILL DETERMINE IF THE TRANSFER IS A DATO OR A DATOB
659 ;AND SET THE CYCLE COUNT APPROXIMATELY VIA R2 AND ALSO SET R3 TO
660 ;THE CORRECT WRITE BUFFER ADDRESS
661 ;*****
662 ;*****
663
664 002016- WREGS:
665 002016- 012702 000300- MOV #WCC, R2 ;SET R2 FOR A DATO CYCLE COUNT
666 002022- 032716 000400 BIT 15, (R5) ;IF BIT 8 OF XFR FUNC IS 0
667 002026- 001412 BCC 1S ;ALL SET 8 OF FINISH UP
668 002030- 033715 001000 BIT 9, (R5) ;IF BIT 9 OF XFR FUNCTION
669 002034- 001407 BCC 1S ;ALL SET 9 OF FINISH UP
670 002036- 012702 000276- MOV #BYTCC, R2 ;ELSE SET R2=ADDR OF BYTCC AND
671 002042- 013737 000300- 000276- MOV WCC, BYTCC ;MOVE VALUE OF WCC TO BYTCC AND
672 002050- 006337 000276- ASL BYTCC ;DOUBLE BY SHIFTING LEFT
673 ;ELSE LEAVE R2 = WCC
674
675 002054- 1S:
676 002062- 063737 000314- 000340- ADD #WBUF, WBFADR ;INC WBFADR ADDR BY VALUE IN WBUF
677 002066- 000207 MOV #WBFADR, R3 ;SET R3 = TO ADDR OF WBFADR
678 ;RETURN
679 ;*****
680 ;*****
681 ;THIS ROUTINE WILL SET UP R3 WITH VALUES FOR DOING READS
682 ;*****
683 ;*****
684
685 002070- RREGS:
686 002070- 012702 000302- MOV #RCC, R2 ;SET R2 FOR A DATI CYCLE COUNT
687 002074- 063737 000315- 000334- ADD #RBUF, RBFADR ;INC RBFADR ADDR BY VALUE IN RBUF
688 002102- 012703 000334- MOV #RBFADR, R3 ;SET R3 = TO VALUE IN RBFADR
689 002106- 000207 RTS PC ;RETURN
690 ;*****
691 ;*****
692 ;*****
693 ;THIS ROUTINE CHECKS FOR A CORRECT DATI TRANSFER SINCE ALL OF THE
694 ;READ TRANSFER BUFFER SIZE CONTAINS THE SAME DATA, ONLY ONE LOCATION
695 ;IS COMPARED WITH THE CONTENTS IN THE DEVICE DATA BUFFER REGISTER.
696 ;*****
697 ;*****
698
699
700 002110- DIDATA:
701 002110- 013703 000334- MOV RBFADR, R3 ;SET R3 = 1ST READ BUFF ADDR
702 002114- 063703 000312- ADD #RBUF, R3 ;GET LAST ADDR BY ADDING BUFF SIZE
703 002120- 013704 000334- MOV RBFADR, R4 ;SET R4=PRESENT READ BUFF ADDR
704
705 002124- 1S:
706 002124- 021024 CMP (R0), (R4)+ ;IS VALUE IN DEV DATA REG=TO THAT IN READ BUFF?
707 002130- 001407 BQO 2S ;IF EQUAL, CONTINUE
708 002136- 005744 MSGNS, BRCIN, DTERR ;ASCII MESSAGE CALL WITH COMMON HEADER
709 002140- 004737 002556- JSR PC, R0DATA ;RETURN R4 TO ERR LOCATION
710 002144- 000402 BR 3S ;AND TYPE OUT CONTENTS OF REGS
711
712 002146- 2S:
713 002146- 020403 CMP R4, R3 ;IS R4 = LAST XFER LOCATION?
```



```

937 002762 052737 000004 000242  BIS #BIT2,MASK ;SET BIT2 TO INDICATE DEV 3 INTERR'D
938 002770 000002  RTI ;RETURN
939 002772 052737 000010 000242  ISR4:  BIS #BIT3,MASK ;SET BIT3 TO INDICATE DEV 4 INTERR'D
940 002772 000002  RTI ;RETURN
941 003006 052737 000020 000242  ISR5:  BIS #BIT4,MASK ;SET BIT4 TO INDICATE DEV 5 INTERR'D
942 003002 000002  RTI ;RETURN
943 003002 052737 000040 000242  ISR6:  BIS #BIT5,MASK ;SET BIT5 TO INDICATE DEV 6 INTERR'D
944 003016 000002  RTI ;RETURN
945 003012 052737 000040 000242  ISR7:  BIS #BIT6,MASK ;SET BIT6 TO INDICATE DEV 7 INTERR'D
946 003020 000002  RTI ;RETURN
947 003022 052737 000100 000242  ISR8:  BIS #BIT7,MASK ;SET BIT7 TO INDICATE DEV 8 INTERR'D
948 003030 000002  RTI ;RETURN
949 003032 052737 000200 000242  ISR9:  BIS #BIT8,MASK ;SET BIT8 TO INDICATE DEV 9 INTERR'D
950 003030 000002  RTI ;RETURN
951 003032 052737 000400 000242  ISR10: BIS #BIT9,MASK ;SET BIT9 TO INDICATE DEV 10 INTERR'D
952 003030 000002  RTI ;RETURN
953 003032 052737 0001000 000242  ISR11: BIS #BIT10,MASK ;SET BIT10 TO INDICATE DEV 11 INTERR'D
954 003030 000002  RTI ;RETURN
955 003032 052737 0004000 000242  ISR12: BIS #BIT11,MASK ;SET BIT11 TO INDICATE DEV 12 INTERR'D
956 003100 000002  RTI ;RETURN
957 003102 003200  NOSET: MMSG3
958 003104 177777  -1
959 003106 003232  ERRSET: MMSG4
960 003110 177777  -1
961 003112 003125  DIERR: MMSG1
962 003114 177777  -1
963 003116 003153  DOERR: MMSG2
964 003120 177777  -1
965 003122 003304  NOINTR: MMSG5
966 003124 177777  -1
967 003126 042045 052101 020111 MMSG1: .ASCIZ "%DATI OR DATIP ERROR"
968 003134 051117 042040 052101
969 003142 050111 042440 051122
970 003150 051117 000 000
971 003152 045 040504 047524 MMSG2: .ASCIZ "%DATO OR DATOB ERROR"
972 003160 047440 020122 040504
973 003166 047524 020102 051105
974 003174 047522 000122
975 003200 042045 053105 041511 MMSG3: .ASCIZ "%DEVICE READY BIT NOT SET"
976 003206 020105 042522 042101
977 003214 020131 044502 020124
978 003222 047516 020124 042523
979 003230 000134 000
980 003232 042045 053105 041511 MMSG4: .ASCIZ "%DEVICE INTERRUPTED ON ERROR--NOT ON DONE"
981 003240 020105 047111 042524
982 003246 051122 050125 042524
  
```

```

993 003252 020104 047117 042440
994 003254 051122 051117 026455
995 003270 047516 020124 047117
996 003276 042040 047117 000105
997 003304 052045 042510 043040 MMSG5: .ASCIZ "%THE FOLLOWING DEVICE(S) DID NOT INTERRUPT:%"
998 003312 046117 047514 044527
999 003320 043516 042040 053105
1000 003326 041511 024105 024523
1001 003334 042040 042111 047040
1002 003342 052117 044440 052116
1003 003350 051105 052522 052120
1004 003356 022472 000
1005 003362 003362  .EVEN
1006 003364 000361  WRTOUT: .RLKW 361
1007 004324 000361  RPEDIN: .RLKW 361
1008
1009 005266  .=.
1010
1011 000001  .FND
  
```

ACSR	000102R	246#	897*																	
ADDR	000006R	212#	396	453	608	858														
ADDR22=	001000	264#																		
ASB	000109R	264#																		
ASTAT	000104R	248#																		
AWAS	000110R	251#																		
BA	000364R	317#	890*	891*	892															
BADHEM	000372R	320#																		
BADR	000374R	320	321#	886*																
BDDATA	002556R	470	477	709	738	872	885#													
BFGIN	000000R	209#	362	366	439	440	445	468	475	512	518	707	736	901						
BT0	000001	264#	411	481	507	589	760	798	931											
BT1	000002	264#	764	934																
BT10	002000	264#	961																	
BT11	004000	264#	964																	
BT12	010000	264#																		
BT13	020000	264#																		
BT14	040000	264#																		
BT15	100000	264#																		
BT2	000004	264#																		
BT3	000010	264#	937																	
BT4	000020	264#	940																	
BT5	000040	264#	758	943																
BT6	000040	264#	762	946																
BT7	000200	264#	943																	
BT8	000200	264#	952																	
BT9	000400	264#	666	955																
BTS	001000	264#	668	958																
BREK3	100400	264#	439	440																
BRK	001070R	438#	444																	
BR1	000012R	214#	545																	
BR2	000013R	215#																		
BRODS	100413R	215#																		
BUFSZ	000250R	278#	553																	
BYTCC	000276R	289#	670	671*	672*															
B12T15	000246R	277#	341																	
CC	003100R	244#	888*	889*	890															
CDDATA	104412R	264#																		
CHEKPK	001742R	409	637#																	
CKMORF	001436R	534#	540	548																
CLRRSE	001466R	347	491	606#																
CONFIC	000056R	234#																		
CR1	000366R	318#	892*	893*	894	896	897													
CR2	000370R	319#	894*	895*																
CSRA	003100R	244#	894*																	
DATAK	001132R	442	449#																	
DATCK	104411	264#																		
DATERS	104400	264#																		
DB	003600R	313#	887*	888																
DEVCNT	000264R	318#	507	550*	551*															
DICNT	000306R	293#	529*	558	591*															
DIDATA	002110R	483	700#																	
DIERR	003120R	707	971#																	
DOCNT	000304R	292#	536*	565	568	594*														
DODATA	002162R	486	729#																	
DDI	000266R	285#	373*	388	410*	411	451*	480*	481	533*	542*	550	589							
DOERR	003116R	736	973#																	

DROP	001404R	345	392	517#																
DV	000240R	274#	324	340	341*	342	343	384	390*	441	450	607	625	861						
DVID1	000014R	216#	332	340																
DVWASK	000400R	321#																		
DVREGS	000300R	311#																		
EABIT	000310R	319#	901																	
EABITS	002234R	371	425	757*	760*	764*														
ENDITS	000336R	365#	756#																	
ENDRBE	000336R	365#	513*																	
ENDWBP	000342R	307#	363*	364*	648															
ENDS	104410	264#	368*																	
ERRITS	000356R	313#	518																	
ERRSET	003106R	413	473																	
ERRTYP	000106R	249#	969#																	
EXINC	001644R	592	899#																	
EXITS	104400	264#	595#																	
GETRUP	001520R	552#																		
GETDEV	002502R	442	855#																	
GETOUT	002554R	866	874#																	
GETPAS	104415	264#	362	366																
GO	001054R	405	433#																	
GWBUPS	104414	264#																		
HRDCNT	000044R	229#																		
HRDEFS	104405	264#																		
HRDPAS	000050R	231#	901																	
ICONT	000036R	226#																		
ICOUNT	000040R	227#																		
IDNWH	000122R	256#																		
IMCCNT	001622R	543	588#																	
INCCDO	001640R	590	593#																	
INIT	000030R	223#																		
INTR	000120R	537																		
ISR1	002142R	255#	331*	338*																
ISR10	003052R	947#	940#																	
ISR11	003062R	960#																		
ISR12	002752R	963#																		
ISR2	002752R	963#																		
ISR3	002762R	963#																		
ISR4	002772R	963#																		
ISR5	003002R	942#																		
ISR6	003002R	945#																		
ISR7	003022R	948#																		
ISR8	003032R	951#																		
ISR9	003032R	954#																		
MAP22S	104412R	264#																		
MASK	000242R	252#																		
MODNAM	000000R	943*	325	342*	384	389	390	435*	441	870	931*	934*	937*	940*						
MODSP	000242R	210#	946*	949*	952*	953*	958*	961*	964*											
MORDEV	000170R	457#	262#																	
MORE	000244R	264#	464	489																
MSCS	104403	264#	445	468	489	403*	450*	458*	535*	607*	6									

MSSG3	003200R	967	985#																	
MSSG4	003232R	965	990#																	
MSSG5	003304R	975	997#																	
NOINTR	003122R	445	975#																	
NOSET	003102R	460	967#																	
NPASS	001320R	461	490#																	
NULL	= 000000	264#																		
NXT	= 001310R	471		478	484	487#														
OPEN	= 000000	211	217	218	219	220	237	238	239	240	241	242	243	244						
		246	248	250	251	253	254	255	264#	298	299	301	302							
OTDAS	= 104420	264#																		
PASCNT	= 000034R	225#																		
PTREQ	= 000004	264#																		
POPSP	= 005726	264#																		
POPSP2	= 000000	264#																		
PRTY	= 000000	264#																		
PRTY0	= 000000	264#																		
PRTY1	= 000040	264#																		
PRTY2	= 000100	264#																		
PRTY3	= 000140	264#																		
PRTY4	= 000200	264#																		
PRTY5	= 000240	264#																		
PRTY6	= 000300	264#																		
PRTY7	= 000340	214	264#																	
PS	= 177776	264#																		
PSW	= 177776	264#																		
PTR1	= 000400R	302#																		
PTR2	= 000402R	325#																		
PUSH	= 005746	264#																		
PUSH2	= 024646	264#																		
P1	= 000226R	269#	376	651																
P2	= 000226R	270#	796	823	838															
P3	= 000230R	271#																		
P4	= 000232R	271#	644	806	808	833	835													
RANDS	= 104417	264#																		
RANNUM	= 000054R	233#																		
RBFAOR	= 000334R	304#	399*	400*	454*	687*	688	701	703	715*	925*									
RBOP	= 000312R	295#	400	560*	687	702	715	825	832	924										
RBUPA	= 000330R	302#	363	399																
RBUPA	= 000326R	301#																		
RBUPAZ	= 000332R	303#																		
RBUPVA	= 000324R	300#	362																	
RCC	= 000302R	291#	563*	564*	686	687	921	922*												
REFDIN	= 004324R	300	494	821	824	828	1007#													
REREST	= 000640R	383#	514																	
RESDAT	= 002402R	394	820#																	
RESRT	= 000000	264#	361#																	
RES1	= 000056R	235#																		
RES2	= 000060R	236#																		
ROTCNT	= 000234R	272#	375*	502	504*	646														
ROTPAT	= 000234R	272#	375*	502	504*	646														
ROTPRO	= 002276R	492	774#																	
RQLVL	= 000354R	312#	378#	427	775	777*	780*	781	783*											
RREGS	= 002070R	413	685#																	
RSTRT	= 000000	264#																		
SAVRI	= 000272R	287#	376*	410*	638															

SAVR5	000274R	288#	377*	411*	493	619															
SBADR	= 000102R	245#																			
SETUP	= 001410R	348	386	528#																	
SIMLGO	= 000336R	339	437*																		
SOPCNT	= 000042R	228#																			
SOPERS	= 104406	264#																			
SOPPAS	= 000046R	230#																			
SPOINT	= 000032R	224#																			
SPSTZ	= 000040	1#																			
SR1	= 000016R	217#																			
SR2	= 000018R	218#																			
SR3	= 000022R	218#																			
SR4	= 000024R	210#																			
START	= 000410R	223																			
STAT	= 000000	272#																			
SVDDDI	= 000270R	286#	388*	451																	
SVRO	= 000067R	237#																			
SVR1	= 000064R	238#																			
SVR2	= 000066R	239#																			
SVR3	= 000070R	240#																			
SVR4	= 000072R	241#																			
SVR5	= 000074R	242#																			
SVR6	= 000076R	243#																			
SWAPCC	= 002704R	509																			
SYSCNT	= 000052R	232#	920#																		
TDCNT	= 000406R	327#	414*	443*																	
TRDFD	= 000022	214#																			
VECTOR	= 000010R	214#	532																		
WASADR	= 000104R	247#																			
WBFAOR	= 000340R	306#	397*	398*	455*	674*	675	730	732	744*											
WBUP	= 000314R	296#	398	573*	674	731	744	925	926*												
WBUPA	= 000322R	296#																			
WBUPA	= 000320R	298#	367	397																	
WBUPVA	= 000316R	297#	366																		
WCC	= 000300R	290#	575*	577*	665	671	922	923*													
WDFR	= 000116R	258#	330*	331*																	
WDT0	= 000114R	253#	329*	336*																	
WREGS	= 002016R	416	664#																		
WRTOUT	= 003360R	297	455	841	844	1006#															
XFLAG	= 000005R	211#																			
XPR1	= 000344R	308#	377	642																	
XPR2	= 000346R	309#																			
XPR3	= 000350R	310#																			
XPR4	= 000352R	311#																			
.	= 005266R	1005#	1006#	1007#	1009#																

- ABS. 000000 000
 005266 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 XBFARO-XREABO/SOL/CRP:SYM=DDXCON,XREABO
 RUN-TIME: 2 3 .4 SECONDS

BEAR DEC/X11 SYSTEM EXERCISER MODULE
XBEABO.P11 12-OCT-78 11:46
RUN-TIME RATIO: 18/5=3.3
CORE USED: 7K (13 PAGES)

MACY11 30A(1052) 12-OCT-78 16:18 PAGE 28
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0026