

.REN I

IDENTIFICATION

PRODUCT CODE: AC-E884C-MC
PRODUCT NAME: CXBEACO M7855 BUS TESTER MODULE
PRODUCT DATE: APRIL 1983
MAINTAINER: DEC/V11 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) 1976,1983 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 UBE(S) DOING DATI(S), DATOR(S), DATIP(S), AND DATO(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 UBE(S): WITH MORE THAN ONE UBE, ALL SHOULD HAVE W1 JUMPER EXCEPT THE UBE 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 UBE(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 UBE(S); WITH MORE THAN ONE UBE, ALL SHOULD HAVE W1 JUMPER EXCEPT THE UBE 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 UBE PROCEEDS IN INCREMENTS OF 20. EVERY BIT OF DVID1 CORRESPONDS TO A PARTICULAR UBE ADDRESS BEING PRESENT; I.E., BIT 0 SET = UBE ADDRESS OF 170000, BIT 1 SET = UBE ADDRESS OF 170020, BIT 2 SET = UBE ADDRESS OF 170040, ETC.

THE UBE(S) CAN BE ON THE BUS IN ANY ADDRESSING SEQUENCE. THE ONLY REQUIREMENT IS THAT DVID1 REFLECTS THE ADDRESSES OF EXISTING UBE(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 (DATI/P OF DATO/B) THE CONTENTS OF THE FOLLOWING REGISTERS PRE-PRINTED AS WELL AS THE MEMORY LOCATION AND CONTENTS. THERE SHOULD BE A DISCREPANCY BETWEEN BEDB AND CONTENTS OF MEMORY ON ERRORS.

BEDB BECC BEBA BECR1 BECR2 MEM-ADDR MEM-CONTENTS

2. FOR "INTERRUPTS ON ERROR - NOT ON DONE" THE ABOVE REGISTERS WILL ALSO BE PRINTED OUT EXCEPT DISREGARD 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 INDICATING WHICH DEVICES INTERRUPTED. BIT COMPARISON OF THESE TWO REGISTERS WILL INDICATE WHICH DEVICES DID OR DID NOT INTERRUPT.

DV MASK

[


```

000102 000000      ACSR:  OPEN      ;CONTENTS OF CSR.
000104      WASADR: ;ADDR OF BAD DATA, OR
000104 000000      ASTAT: OPEN      ;STATUS REG CONTENTS.
000106      ERR1YP: ;TYPE OF ERROR
000106 000000      ASB:   OPEN      ;EXPECTED DATA.
000110 000000      AWAS:  OPEN      ;ACTUAL DATA.
000112 000534      RSTR:  RESTR     ;RESTART ADDRESS AFTER END OF PASS
000114 000000      WDT0:  OPEN      ;WORDS TO MEMORY PER ITERATION
000116 000000      WDFR:  OPEN      ;WORDS FROM MEMORY PER ITERATION
000120 000000      INTR:  OPEN      ;# OF INTERRUPTS PER ITERATION
000122 000073      IDNUM: 73        ;MODULE IDENTIFICATION NUMBER=73
      000040      .REPT  SPSIZ      ;MODULE STACK STARTS HERE.
      .NLIST
      .WORD  0
      .LIST
      .ENDR

000224      MODSP:
;*****

```

```

232 .ENABL AMA
233
251
252 000224 000000 P1: 0 ;1ST DATA PATTERN
253 000226 137777 P2: 137777 ;2ND DATA PATTERN
254 000230 000000 P3: 0 ;3RD DATA PATTERN
255 000232 100000 P4: 100000 ;4TH DATA PATTERN
256 000234 000000 ROTCNT: 0 ;COUNTER FOR # OF TIMES DATA ROTATED
257 000236 170014 SIMLGO: 170014 ;ADDR TO SET OFF ALL DEVS SIMULTANEOUSLY
258 000240 000000 DV: 0 ;TEMP STORAGE FOR DVID1
259 000242 000000 MASK: 0 ;USED TO DROP ANY DEV NOT INTERRUPTING
260 000244 000000 MORE: 0 ;WORKING STORAGE FOR DVID1
261 000246 170000 B12T15: 170000 ;USED TO CLEAR BITS 12 THRU 15
262 000250 000740 BUFSZ: 740 ;SIZE FOR 1 DEV
263 000252 000360 360 ;SIZE FOR 2 DEVS
264 000254 000240 240 ;SIZE FOR 3 DEVS
265 000256 000170 170 ;SIZE FOR 4 DEVS
266 000260 000140 140 ;SIZE FOR 5 DEVS
267 000262 000120 120 ;SIZE FOR 6 DEVS
268 000264 000000 DEVCNT: 0 ;TOTAL NUMBER OF DEVS
269 000266 000000 DODI: 0 ;DATA OR DATI INDICATOR
270 000270 000000 SVDODI: 0 ;STORAGE FOR DODI
271 000272 000000 SAVR1: 0 ;LOC TO SAVE R1
272 000274 000000 SAVR5: 0 ;LOC TO SAVE R5
273 000276 000000 BYTCC: 0 ;CYCLE COUNT FOR A BYTE TRANSFER
274 000300 000000 WCC: 0 ;CYCLE COUNT FOR A DATO TRANSFER
275 000302 000000 RCC: 0 ;CYCLE COUNT FOR A DATI TRANSFER
276 000304 000000 DOCNT: 0 ;# OF DEVS DOING A DATO/B(S)
277 000306 000000 DICNT: 0 ;# OF DEVS DOING A DATI/P(S)
278 000310 000000 EABIT: 0 ;USED TO SET EXTENDED MEMORY ADDR BITS
279 000312 000000 RBUF: 0 ;SIZE OF BUFFER FOR DATI/P(S)
280 000314 000000 WBUF: 0 ;SIZE OF BUFFER FOR DATO/B(S)
281 000316 003444 WBUFVA: WRTOUT ;VIRTUAL ADDR OF WRITE BUFFER
282 000320 000000 WBUFPA: OPEN ;PHYSICAL ADDR OF WRITE BUFFER
283 000322 000000 WBUFEA: OPEN ;EXT MEM ADDR BITS SET BY MONITOR
284 000324 004406 RBUFVA: REEDIN ;VIRTUAL ADDR OF READ BUFFER
285 000326 000000 RBUFPA: OPEN ;PHYSICAL ADDR OF READ BUFFER
286 000330 000000 RBUFEA: OPEN ;EXT MEM ADDR BITS SET BY MONITOR
287 000332 000360 RBUFSZ: 360 ;TOTAL SIZE OF BUFFER RESERVED FOR READS
288 000334 000000 RBFADR: 0 ;USED TO POINT TO LOC WITHIN RBUFSZ
289 000336 000000 ENDRBF: 0 ;USED FOR LAST ADDR(PHYSICAL) OF READ BUFF
290 000340 000000 WBFADR: 0 ;USED TO POINT TO LOC WITHIN WBUFSZ
291 000342 000000 ENDWBF: 0 ;USED FOR LAST ADDR(PHYSICAL) OF WRITE BUFF
292 000344 004100 XFR1: 4100 ;FUN 2-DATI-INTR
293 000346 005500 XFR2: 5500 ;FUN 2-DATOB-INTR
294 000350 022500 XFR3: 22500 ;FUN 1-DATI/NO ROT-INTR
295 000352 003100 XFR4: 3100 ;FUN 1-DATO-INTR
296 000354 000060 RQLVL: 60 ;USED TO SET REQUEST LEVELS OF DEVS
297 000356 007740 ERBITS: 7740 ;USED TO CHECK IF ANY ERROR BITS SET
298 000360 DVREGS:
299 000360 000000 DB: 0 ;DEV DATA REG CONTENTS
300 000362 000000 CC: 0 ;DEV CYCLE COUNT CONTENTS
301 000364 000000 BA: 0 ;DEV ADDR REG CONTENTS
302 000366 000000 CR1: 0 ;DEV CR1 REG CONTENTS
303 000370 000000 CR2: 0 ;DEV CR2 REG CONTENTS
304 000372 000774 BADMEM: BADR ;BAD MEMORY ADDR
305 000374 000000 BADR: 0 ;CONTENTS OF BAD MEMORY

```


306 000376 177777 177777
307 000400 DVMASK: ;THIS POINTER IS USED SO THE NEXT
308 000400 000240' PT1: DV ;TWO LOCATIONS WILL BE PRINTED OUT FOR
309 000402 000242' PT2: MASK ;AN ERR WHICH DEVICES DID NOT INTERRUPT
310 000404 177777 -1
311 000406 000000 TJCNT: 0 ;TIME OUT COUNTER
312
313 000410 012737 000361 000114' TART: MOV #361,WDT0 ;AT LEAST 361 WORDS TO MEM/ITERATION
314 000416 012737 000361 000116' MOV #361,WDFR ;AT LEAST 361 WORDS FROM MEM/ITERATION
315 000424 012737 000001 000120' MOV #1,INTR ;AT LEAST 1 INTERRUPT /ITERATION
316 000432 013700 000014' MOV DVID1,RO ;SAVE DEV COUNT
317 000436 006200 2\$: ASR RO ;SHIFT IN A COUNT
318 000440 001412 BEQ 3\$;BR OUT IF NONE LEFT
319 000442 103375 BCC 2\$;BR BACK IF NO BIT IN THIS POSITION
320 000444 062737 000361 000114' ADD #361,WDT0 ;361 MORE WORDS TO MEM
321 000452 062737 000361 000116' ADD #361,WDFR ;361 MORE WORDS FROM MEM
322 000460 005237 000120' INC INTR ;1 MORE INTERRUPT
323 000464 000764 BR 2\$;GO CHECK FOR MORE
324 000466 013737 000014' 000240' 3\$: MOV DVID1,DV ;MOVE DVID1 TO TEMP STORAGE
325 000474 043737 000246' 000240' BIC B12T15,DV ;ENSURE BITS 12-15 ARE CLEARED
326 000502 013737 000240' 000242' MOV DV,MASK ;INIT MASK BITS EQUAL TO DVID1
327 000510 013737 000240' 000244' MOV DV,MORE ;THEN SET UP WORKING AREA
328 000516 001002 BNE 1\$;GO TO 1\$ IF THERE ARE DEVS
329 000520 000137 001404' JMP DROP ;ELSE DROP THE MODULE
330 000524 1\$:
331 000524 004737 001646' JSR PC,CLRREG ;CLEAR ALL DEV REGS
332 000530 004737 001410' JSR PC,SETUP ;SET UP AVAILABLE VEC LOCATIONS AND
333 ;GET BUFFER SIZES FOR DATO(S) & DATI(S)
334
335 ;*****
336 ;*****
337 ;NOTE: IN THE FOLLOWING COMMENTS THE TERM READ IS USED
338 ;TO DENOTE DATI OR DATIP AND THE TERM WRITE DENOTES
339 ;DATO OR DATOB.
340 ;*****
341 ;*****
342 ;THIS IS THE LOCATION THE PROGRAM WILL PROCEED
343 ;WITH AFTER AN END OF PASS
343 000534 RESTRT:
344 000534 104415 000000' 000524' GETPA\$,BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
345 000542 013737 000326' 000336' MOV RBUFPA,ENDRBF ;MOVE READ BUFF ADDR TO END-OF-READ BUFF REG
346 000550 062737 000736 000336' ADD #736,ENDRBF ;ADD 736 TO GET LAST ADDR OF BUFFER
347
348 000556 104415 000000' 000316' GETPA\$,BEGIN, WBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT WBUFVA
349 000564 013737 000320' 000342' MOV WBUFPA,ENDWBF ;MOVE WRITE BUFF ADDR TO END-OF-WRITE BUFF REG
350 000572 062737 000736 000342' ADD #736,ENDWBF ;ADD 736 TO GET LAST ADDR OF BUFFER
351
352
353 ;RSOQ1 JSR PC,FABITS ;SET BITS 0 & 1 OF REA EQUAL TO
354 ;BITS 4 & 5 OF RBUFEA RESPECTIVELY
355 000600 012737 000001 000266' MOV #1,DODI ;INIT XFER INDICATOR TO 1 SO THE
356 ;1ST INC WILL SIGNIFY A DATI
357 000606 005037 000234' CLR ROTCNT ;CLEAR PATTERN ROTATION COUNT
358 000612 012737 000224' 000272' MOV #P1,SAVR1 ;INIT SAVR1 TO 1ST PATTERN
359 000620 012737 000344' 000274' MOV #XFR1,SAVR5 ;INIT SAVR5 TO 1ST XFER FUNCTION
360 000626 012737 000060 000354' MOV #60,RQLVL ;INIT REQUEST LEVEL TO NPR/INTR BR?

```

361
362
363 ;THIS IS THE LOCATION THE PROGRAM WILL PROCEED WITH
364 ;AFTER AN ITERATION
365 000634
366 000634 023737 000242' 000240' REREST:
367 000642 001402 CMP MASK,DV ;# OF DEVS THAT SHOULD INTR=# OF DEVS THAT DID?
368 000644 004737 001410' BEQ 1$ ;IF EQUAL,GO TO 2$
369 000650 JSR PC,SETUP ;ELSE GET NEW DEV COUNTS & BUFFER SIZES
370 000650 013737 000266' 000270' 1$: MOV DODI,SVDODI ;SAVE DODI SETTING
371 000656 013737 000242' 000244' MOV MASK,MORE ;RESET MORE BITS
372 000664 013737 000242' 000240' MOV MASK,DV ;IF ANY DEVS DROPPED, ALTER DV
373 000672 001002 BNE 2$ ;IF NOT, CONTINUE ON
374 000674 000137 001404' JMP DROP ;ELSE DROP MODULE
375 000700
376 000700 004737 002464' 2$: JSR PC,RESDAT ;RESTORE DATA IN READ BUFFER
377 ;AND CLEAR WRITE BUFFER
378 000704 013700 000006' MOV ADDR,R0 ;PUT DEVICE ADR INTO REG. 0
379 000710 013737 000320' 00340' MOV WBUFPA,WBFADR ;USE WBFADR AS WRITE BUFF ADDR
380 000716 13737 000314' 000340' SUB WBUF,WBFADR ;INIT TO WRITE BUFF LESS WBUF
381 000724 13737 000326' 000334' MOV RBUFPA,RBFADR ;USE RBFADR AS READ BUFF ADDR
382 000732 13737 000312' 000334' SUB RBUF,RBFADR ;INIT TO READ BUFF LESS RBUF
383
384 000740 3$:
385 000740 006137 000244' ASR MORE ;CHECK FOR ANOTHER DEV
386 000744 10340' RCS 4$ ;IF THERE, CONTINUE
387 000746 001442 BQ GO ;AND IF NOT, GO TO GO
388 000750 062700 000020 A J #20,R0 ;ELSE ADD 20 FOR NEXT ADDR
389 000754 000771 BH 3$ ;AND CHECK FOR MORE
390 000756
391 000756 004737 001742' 4$: JSR PC,CHEKPX ;SEE IF R1 & R5 NEED INITIALIZATION
392 000762 005237 000266' INC DODI ;INC XFER INDICATOR
393 000766 032737 000001 000266' BIT #BIT0,DODI ;IF BIT 0 OF DODI IS 1
394 000774 001003 BNE 5$ ;GO SET UP REGS FOR DATO(S)
395 000776 004737 002070' JSR PC,RREGS ;ELSE SET UP REGS FOR DATI(S)
396 001002 000402 BR 6$ ;GO LOAD THE REGISTERS
397 001004
398 001004 004737 002016' 5$: JSR PC,WREGS ;SET REGS FOR WRITE XFER INFO
399
400 ;AT THIS POINT THE DEVICE REGISTERS WILL BE LOADED TO DO
401 ;TRANSFERS
402
403 001010 6$:
404 001010 004737 002234' JSR PC,EABITS ;SET EXTENDED ADDRESS BITS IN EABIT ;RS001
405 001014 012120 MOV (R1), (R0) ;SET UP DATA BUFFER REG
406 001016 011220 MOV (R2), (R0) ;SET UP CYCLE COUNT REG.
407 001020 011320 MOV (R3), (R0) ;SET UP ADDR REG
408 001022 013760 000310' 000010 MOV EABIT,10(R0) ;SET UP EXT ADDR BITS IN CR2
409 001030 012510 MOV (R5), (R0) ;SET UP CR1
410 001032 053710 000354' BIS RQLVL,(R0) ;SET REQUEST LEVELS
411 001036 062700 000012' ADD #12,R0 ;ADD 12 TO GET NEXT DEV ADDR
412
413 001042 010137 000272' MOV R1,SAVR1 ;SAVE CONTENTS OF R1(LAST DATA PATTERN)
414 001046 010537 000274' MOV R5,SAVR5 ;SAVE CONTENTS OF R5(LAST XFER INSTR.)
415 001052 000732 BR 3$ ;GO LOAD ANOTHER DEVICE
416 001054
417 001054 005037 000406' GO: CLR TOCNT ;CLEAR TIME OUT COUNT

```

```

418 001060 005037 000242' CLR MASK ;CLEAR ALL BITS IN MASK
419 ;TO BE RESET IN INTR SERV RTNS
420 001064 005277 1.77146 INC @SIMLGO ;SET OFF ALL DEVS
421 001070 BRK:
422 001070 104407 000000' BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
001074 104407 000000' ;BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
423 001100 023737 000240' 000242' CMP DV,MASK ;ALL DEVS INTR?
424 001106 001411 BEQ DATAACK ;IF YES, GO TO DATAACK
425 001110 005337 000406' DEC TOCNT ;ELSE DEC TIME OUT COUNT
426 001114 001365 BNE BRK ;AND BREAK IF NOT 0
427 001116 104403 000000' 003204' MSGN$,BEGIN,NOINTR ;ASCII MESSAGE CALL WITH COMMON HEADER
428 001124 005004 CLR R4 ;R4 INVALID IN ERR REPORT
429 001126 004737 002564' JSR PC,GETDEV ;REPORT NON-INTR'G DEVICES
430
431 001132 DATAACK:
432 001132 013737 000240' 000244' MOV DV,MORE ;SET UP MORE
433 001140 013737 000270' 000266' MOV SVDODI,DODI ;RESTORE DODI TO SETTING WHEN
434 ;LOADING 1ST DEV
435 001146 013700 000006' MOV ADDR,R0 ;INIT R0 TO 1ST DEV ADDR
436 001152 012737 004406' 000334' MOV @REEDIN,RBFADR ;RBFADR=1ST LOC IN READ BUFFER
437 001160 012737 003444' 000340' MOV @WRTOUT,WBFADR ;WBFADR=1ST LOC IN WRITE BFFER
438 001166 005003 CLR R3 ;USED IN DODATA RTN
439 001170 MORDEV:
440 001170 006237 000244' ASR MORE ;CHECK FOR OTHER DEVS
441 001174 103405 BCS 2$ ;IF THERE CONTINUE
442 001176 001001 BNE 1$ ;IF SOME LEFT,GO CHECK FOR MORE
443 001200 000447 BR NPASS ;ELSE GO TO END OF PASS RTN
444 001202 1$:
445 001202 062700 000020 ADD @20,R0 ;GET NEXT DEV ADDR
446 001206 000770 BR MORDEV ;GO CHECK FOR THAT DEV
447 001210 2$:
448 001210 105760 000006 TSTB 6(R0) ;IS READY BIT SET IN CR1?
449 001214 100407 BMI 3$ ;IF SET, CONTINUE
450 001216 104403 000000' 003164' MSGN$,BEGIN,NOSET ;ASCII MESSAGE CALL WITH COMMON HEADER
451 001224 005004 CLR R4 ;R4 INVALID IN ERR REPORT
452 001226 004737 JSR PC,BDDATA ;ELSE TYPE OUT CONTENTS OF ALL DEV REGS
453 001232 000425 BR NXT ;AND GO CHECK FOR NEXT DEV
454 001234 3$:
455 001234 033760 000356' 000016 BIT ERRBITS,16(R0) ;SEE IF ANY ERROR BITS SET
456 001242 001407 BEQ 4$ ;IF NONE SET, CONTINUE
457 001244 104403 000000' 003170' MSGN$,BEGIN,ERRSET ;ASCII MESSAGE CALL WITH COMMON HEADER
458 001252 005004 CLR R4 ;R4 INVALID IN ERR REPORT
459 001254 004737 JSR PC,BDDATA ;ELSE TYPE OUT CONTENTS OF ALL DEV REGS
460 001260 000413 BR NXT ;AND GO CHECK FOR NEXT DEV
461 001262 4$:
462 001262 005237 000266' INC DODI ;INC XFER INDICATOR
463 001266 032737 000001' 000266' BIT @BIT0,DODI ;DATI OR DATO ?
464 001274 001003 BNF 5$ ;IF DATO,GO DO IT
465 001276 004737 002110' JSR PC,DIDATA ;CHECK DATI XFERS
466 001302 000402 BR NXT ;CHECK FOR NEXT DEV
467 001304 5$:
468 001304 004737 002162' JSR PC,DODATA ;CHECK DATO XFERS
469 001310 NXT:
470 001310 062700 000020 ADD @20,R0 ;ADD 20 FOR NEXT DEV ADDR
471 001314 000137 001170' JMP MORDEV ;GO CHECK FOR ANOTHER DEV
472 001320 NPASS:
473 001320 004737 001646' JSR PC,CLRREG ;CLEAR ALL DEV REGS
    
```

```

474 001324 004737 00236J'      JSR    PC,ROTRQS      ;SHIFT REQUEST TO NEXT BR LEVEL
475 001330 022737 000352' 000274'  CMP    0XFR4,SAVR5   ;HAS LAST XFER BEEN DONE?
476 001336 001010                BNE    1$            ;IF NOT, CONTINUE
477                                ;*****
                                ;*****
478                                ;AFTER P2 & P4 HAVE BEEN ROTATED 16 TIMES, ALLOW THE POINTER
479                                ;R1 TO INCREMENT THRU THE END OF THE READ BUFFER BUT DON'T ALLOW ROTATION
480                                ;*****
                                ;*****
481                                ;*****
482 001340 022737 000020 000234'  CMP    016.,ROTCNT   ;ELSE HAS P2&P4 BEEN ROTATED 16 TIMES?
483 001346 001404                BEQ    1$            ;IF YES DON'T ROTATE PATTERNS ANYMORE
484 001350 005337 000234'      INC    ROTCNT        ;ELSE INC ROTATION COUNT, AND
485 001354 004737 002424'      JSR    PC,ROTPAT     ;ROTATE DATA PATTERNS
486 001360                1$:
487 001360 032737 000001 000264'  BIT    0BIT0,DEVCNT ;IS THERE AN ODD # OF DEVS?
488 001366 001402                BEQ    2$            ;IF NOT,GO TO 5$
489 001370 004737 003766'      JSR    PC,SWAPCC     ;ELSE SWAP RCC WITH WCC & RBUF
490                                ;WITH WBUF FOR CORRECT XFER INFO
491                                2$:
492 001374 001374 104413 000000'  ENDIT$,BEGIN        ;SIGNAL END OF ITERATION.
493                                ;MONITOR SHALL TEST END OF PASS
494                                ;*****
495 001400 000137 000034'      JMP    REREST        ;ELSE GO TO REREST
496                                ;*****
497                                ;*****
498                                ;*****
499                                ;*****
500                                ;THIS ROUTINE WILL SET UP THE DEVICE INTO VECTORS, COUNT THE
501                                ;NUMBER OF DEVICES, SET THE TRANSFER SIZE OF THE READ AND WRITE
502                                ;BUFFERS, AND SET UP THE READ AND WRITE CYCLE COUNTS.
503                                ;*****
504                                ;*****
505                                ;*****
506                                ;*****
507                                ;*****
508                                ;*****
509                                ;*****
510                                ;*****
511                                ;*****
512                                ;*****
513                                ;*****
514                                ;*****
515                                ;*****
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                                ;*****
572                                ;*****
573                                ;*****
574                                ;*****
575                                ;*****
576                                ;*****
577                                ;*****
578                                ;*****
579                                ;*****
580                                ;*****
581                                ;*****
582                                ;*****
583                                ;*****
584                                ;*****
585                                ;*****
586                                ;*****
587                                ;*****
588                                ;*****
589                                ;*****
590                                ;*****
591                                ;*****
592                                ;*****
593                                ;*****
594                                ;*****
595                                ;*****
596                                ;*****
597                                ;*****
598                                ;*****
599                                ;*****
600                                ;*****
601                                ;*****
602                                ;*****
603                                ;*****
604                                ;*****
605                                ;*****
606                                ;*****
607                                ;*****
608                                ;*****
609                                ;*****
610                                ;*****
611                                ;*****
612                                ;*****
613                                ;*****
614                                ;*****
615                                ;*****
616                                ;*****
617                                ;*****
618                                ;*****
619                                ;*****
620                                ;*****
621                                ;*****
622                                ;*****
623                                ;*****
624                                ;*****
625                                ;*****
626                                ;*****
627                                ;*****
628                                ;*****
629                                ;*****
630                                ;*****
631                                ;*****
632                                ;*****
633                                ;*****
634                                ;*****
635                                ;*****
636                                ;*****
637                                ;*****
638                                ;*****
639                                ;*****
640                                ;*****
641                                ;*****
642                                ;*****
643                                ;*****
644                                ;*****
645                                ;*****
646                                ;*****
647                                ;*****
648                                ;*****
649                                ;*****
650                                ;*****
651                                ;*****
652                                ;*****
653                                ;*****
654                                ;*****
655                                ;*****
656                                ;*****
657                                ;*****
658                                ;*****
659                                ;*****
660                                ;*****
661                                ;*****
662                                ;*****
663                                ;*****
664                                ;*****
665                                ;*****
666                                ;*****
667                                ;*****
668                                ;*****
669                                ;*****
670                                ;*****
671                                ;*****
672                                ;*****
673                                ;*****
674                                ;*****
675                                ;*****
676                                ;*****
677                                ;*****
678                                ;*****
679                                ;*****
680                                ;*****
681                                ;*****
682                                ;*****
683                                ;*****
684                                ;*****
685                                ;*****
686                                ;*****
687                                ;*****
688                                ;*****
689                                ;*****
690                                ;*****
691                                ;*****
692                                ;*****
693                                ;*****
694                                ;*****
695                                ;*****
696                                ;*****
697                                ;*****
698                                ;*****
699                                ;*****
700                                ;*****
701                                ;*****
702                                ;*****
703                                ;*****
704                                ;*****
705                                ;*****
706                                ;*****
707                                ;*****
708                                ;*****
709                                ;*****
710                                ;*****
711                                ;*****
712                                ;*****
713                                ;*****
714                                ;*****
715                                ;*****
716                                ;*****
717                                ;*****
718                                ;*****
719                                ;*****
720                                ;*****
721                                ;*****
722                                ;*****
723                                ;*****
724                                ;*****
725                                ;*****
726                                ;*****
727                                ;*****
728                                ;*****
729                                ;*****
730                                ;*****
731                                ;*****
732                                ;*****
733                                ;*****
734                                ;*****
735                                ;*****
736                                ;*****
737                                ;*****
738                                ;*****
739                                ;*****
740                                ;*****
741                                ;*****
742                                ;*****
743                                ;*****
744                                ;*****
745                                ;*****
746                                ;*****
747                                ;*****
748                                ;*****
749                                ;*****
750                                ;*****
751                                ;*****
752                                ;*****
753                                ;*****
754                                ;*****
755                                ;*****
756                                ;*****
757                                ;*****
758                                ;*****
759                                ;*****
760                                ;*****
761                                ;*****
762                                ;*****
763                                ;*****
764                                ;*****
765                                ;*****
766                                ;*****
767                                ;*****
768                                ;*****
769                                ;*****
770                                ;*****
771                                ;*****
772                                ;*****
773                                ;*****
774                                ;*****
775                                ;*****
776                                ;*****
777                                ;*****
778                                ;*****
779                                ;*****
780                                ;*****
781                                ;*****
782                                ;*****
783                                ;*****
784                                ;*****
785                                ;*****
786                                ;*****
787                                ;*****
788                                ;*****
789                                ;*****
790                                ;*****
791                                ;*****
792                                ;*****
793                                ;*****
794                                ;*****
795                                ;*****
796                                ;*****
797                                ;*****
798                                ;*****
799                                ;*****
800                                ;*****
801                                ;*****
802                                ;*****
803                                ;*****
804                                ;*****
805                                ;*****
806                                ;*****
807                                ;*****
808                                ;*****
809                                ;*****
810                                ;*****
811                                ;*****
812                                ;*****
813                                ;*****
814                                ;*****
815                                ;*****
816                                ;*****
817                                ;*****
818                                ;*****
819                                ;*****
820                                ;*****
821                                ;*****
822                                ;*****
823                                ;*****
824                                ;*****
825                                ;*****
826                                ;*****
827                                ;*****
828                                ;*****
829                                ;*****
830                                ;*****
831                                ;*****
832                                ;*****
833                                ;*****
834                                ;*****
835                                ;*****
836                                ;*****
837                                ;*****
838                                ;*****
839                                ;*****
840                                ;*****
841                                ;*****
842                                ;*****
843                                ;*****
844                                ;*****
845                                ;*****
846                                ;*****
847                                ;*****
848                                ;*****
849                                ;*****
850                                ;*****
851                                ;*****
852                                ;*****
853                                ;*****
854                                ;*****
855                                ;*****
856                                ;*****
857                                ;*****
858                                ;*****
859                                ;*****
860                                ;*****
861                                ;*****
862                                ;*****
863                                ;*****
864                                ;*****
865                                ;*****
866                                ;*****
867                                ;*****
868                                ;*****
869                                ;*****
870                                ;*****
871                                ;*****
872                                ;*****
873                                ;*****
874                                ;*****
875                                ;*****
876                                ;*****
877                                ;*****
878                                ;*****
879                                ;*****
880                                ;*****
881                                ;*****
882                                ;*****
883                                ;*****
884                                ;*****
885                                ;*****
886                                ;*****
887                                ;*****
888                                ;*****
889                                ;*****
890                                ;*****
891                                ;*****
892                                ;*****
893                                ;*****
894                                ;*****
895                                ;*****
896                                ;*****
897                                ;*****
898                                ;*****
899                                ;*****
900                                ;*****
901                                ;*****
902                                ;*****
903                                ;*****
904                                ;*****
905                                ;*****
906                                ;*****
907                                ;*****
908                                ;*****
909                                ;*****
910                                ;*****
911                                ;*****
912                                ;*****
913                                ;*****
914                                ;*****
915                                ;*****
916                                ;*****
917                                ;*****
918                                ;*****
919                                ;*****
920                                ;*****
921                                ;*****
922                                ;*****
923                                ;*****
924                                ;*****
925                                ;*****
926                                ;*****
927                                ;*****
928                                ;*****
929                                ;*****
930                                ;*****
931                                ;*****
932                                ;*****
933                                ;*****
934                                ;*****
935                                ;*****
936                                ;*****
937                                ;*****
938                                ;*****
939                                ;*****
940                                ;*****
941                                ;*****
942                                ;*****
943                                ;*****
944                                ;*****
945                                ;*****
946                                ;*****
947                                ;*****
948                                ;*****
949                                ;*****
950                                ;*****
951                                ;*****
952                                ;*****
953                                ;*****
954                                ;*****
955                                ;*****
956                                ;*****
957                                ;*****
958                                ;*****
959                                ;*****
960                                ;*****
961                                ;*****
962                                ;*****
963                                ;*****
964                                ;*****
965                                ;*****
966                                ;*****
967                                ;*****
968                                ;*****
969                                ;*****
970                                ;*****
971                                ;*****
972                                ;*****
973                                ;*****
974                                ;*****
975                                ;*****
976                                ;*****
977                                ;*****
978                                ;*****
979                                ;*****
980                                ;*****
981                                ;*****
982                                ;*****
983                                ;*****
984                                ;*****
985                                ;*****
986                                ;*****
987                                ;*****
988                                ;*****
989                                ;*****
990                                ;*****
991                                ;*****
992                                ;*****
993                                ;*****
994                                ;*****
995                                ;*****
996                                ;*****
997                                ;*****
998                                ;*****
999                                ;*****
1000                               ;*****

```

```

525 001506
526 001506 013737 000266' 000264' 2$:      MOV      DODI,DEV CNT      ;DEV XFER INDICATOR HAS # OF DEVS
527 001514 005337 000264'          DEC      DEV CNT          ;+ 1, SO DEC DEV CNT BY 1
528 001520          GFIBUF:
529 001520 012700 000246'          MOV      #BUFSZ-2,R0      ;SET R0 TO SIZE TABLE
530 001524 005001          CLR      R1              ;CLEAR R1
531 001526          1$:
532 001526 005720          TST      (R0)+           ;MOVE POINTER TO NEXT VALUE
533 001530 005201          INC      R1              ;INC COUNTER
534 001532 020137 000306'          CMP      R1,DICNT       ;R1=# OF DATI DEVS
535 001536 103773          BLO      1$             ;IF NOT,INC COUNTER & POINTER
536 001540 011037 000312'          MOV      (R0),RBUF      ;ELSE R0 HAS SIZE OF READ BUFF PER DEV
537                                ;1 WORD * 2 LOCATIONS(OR BYTES)
538 001544 011037 000302'          MOV      (R0),RCC       ;AND ALSO IS DATI XFER COUNT
539 001550 006237 000302'          ASR      RCC            ;WHEN IT IS HALVED
540 001554 005437 000302'          NEG      RCC            ;GET 2'S COMP FOR DEV CC REG
541 001560 005737 000304'          TST      DOCNT         ;IF DOCNT IS 0
542 001564 001405          BEQ      2$             ;GO USE THE VALUE R0 POINTS TO
543
544 001566 020137 000304'          CMP      R1,DOCNT      ;R1=# OF DATO DEVS ?
545 001572 001402          BEQ      2$             ;YES, CONTINUE
546 001574 005301          DEC      R1            ;NO, DEC R1
547 001576 005740          TST      -(R0)         ;AND DEC POINTER
548 001600
549 001600 011037 000314' 2$:      MOV      (R0),WBUF      ;R0 HAS SIZE OF WRITE BUFF PER DEV
550                                ;1 WORD * 2 LOCATIONS(OR BYTES)
551 001604 011037 000300'          MOV      (R0),WCC       ;AND ALSO IS DATO XFER COUNT
552 001610 006237 000300'          ASR      WCC            ;WHEN IT IS HALVED
553 001614 005437 000300'          NEG      WCC            ;GET 2'S COMP FOR DEV CC REG
554 001620 000207          RTS      PC            ;RETURN
555
556
557
558
559
560
561
562 001622          INCCNT:
563 001622 032737 000001 000266'      BIT      #BIT0,DODI     ;IF BIT 0 IS 1
564 001630 001003          BNE      INCDO         ;GO INC DOCNT
565 001632 005237 000306'          INC      DICNT         ;ELSE INC DICNT
566 001636 000402          BR      EXINC          ;AND EXIT RTN
567 001640          INCDO:
568 001640 005237 000304'          INC      DOCNT         ;INC DOCNT
569 001644          EXINC:
570 001644 000207          RTS      PC            ;RETURN
571
572
573
574
575
576
577

```

```

578 001646          CLRREG:
579 001646 013737 000240' 000244'  MOV    DV,MORE      ;RESTORE MORE
580 001654 013701 000006'          MOV    ADDR,R1      ;INIT R1 TO 1ST DEV ADDR
581 001660          1$:
582 001660 006237 000244'          ASR    MORE          ;IS THERE A DEVICE?
583 001664 103404          BCS    2$           ;IF YES,GO SET IT UP
584 001666 001421          BEQ    3$           ;IF NONE LEFT, - GET OUT
585 001670 062701 000020          ADD    @20,R1       ;ELSE INC R1 TO NEXT DEV ADDR
586 001674 000771          BR     1$           ;AND GO SEE IF IT'S THERE
587 001676          2$:
588 001676 005011          CLR    (R1)         ;CLEAR DATA BUFFER REG
589 001700 005061 000002          CLR    2(R1)       ;CLEAR CYCLE COUNT REG
590 001704 005061 000004          CLR    4(R1)       ;CLEAR BUFFER ADDR REG
591 001710 005061 000006          CLR    6(R1)       ;CLEAR CR1 REG
592 001714 005061 000010          CLR    10(R1)      ;CLEAR ERROR CLEAR REG
593 001720 005061 000016          CLR    16(R1)      ;CLEAR CR2 REG
594 001724 062701 000020          ADD    @20,R1       ;INC R1 TO NEXT DEV ADDR
595 001730 000753          BR     1$           ;GO CHECK FOR ANOTHER DEV
596 001732          3$:
597 001732 013737 000240' 000244'  MOV    DV,MORE      ;RESTORE MORE
598 001740 000207          RTS    PC           ;RETURN
599
600
601
602
603
604
605
606
607 001742          CHECKPX:
608 001742 013701 000272'          MOV    SAVR1,R1     ;RESTORE R1 TO PATTERN IN SAVR1
609 001746 013705 000274'          MOV    SAVR5,R5     ;RESTORE R5 TO XFR FUNCTION IN SAVR5
610 001752 022705 000352'          CMP    @XFR4,R5     ;DOES R5 POINT TO LAST XFER?
611 001756 100002          BPL    1$           ;IF NOT, CONTINUE
612 001760 012705 000344'          MOV    @XFR1,R5     ;ELSE INIT TO 1ST XFER
613 001764          1$:
614 001764 022701 000232'          CMP    @P4,R1       ;IF R1 DOES NOT EXCEED LAST PATTERN
615 001770 100011          BPL    3$           ;ADDRESS, EXIT ROUTINE
616 001772 022737 000020 000234'  CMP    @16.,ROTCNT  ;HAVE P2&P4 BEEN ROTATED 16 TIMES?
617 002000 001003          BNE    2$           ;IF NOT,INIT R1
618 002002 023701 000336'          CMF    ENDRBF,R1    ;IS R1 AT LAST ADDR OF READ BUFF?
619 002006 100002          BPL    3$           ;IF LESS,GO TO EXINIT
620 002010          2$:
621 002010 012701 000224'          MOV    @P1,R1       ;ELSE INIT R1 TO 1ST DATA PATTERN
622 002014          3$:
623 002014 000207          RTS    PC           ;RETURN
624
625
626
627
628
629
630
;*****
;*****
;THIS ROUTINE WILL DETERMINE IF THE TRANSFER IS A DATO OR A DATOB
;AND SET THE CYCLE COUNT APPROXIMATELY VIA R2 AND ALSO SET R3 TO
;THE CORRECT WRITE BUFFER ADDRES
;*****
;*****

```

```

631
632 002016          WREGS:
633 002016 012702 000300'  MOV    #WCC,R2      ;SET R2 FOR A DATO CYCLE COUNT
634 002022 032715 000400'  BIT    #BIT8,(R5)   ;IF BIT 8 OF XFER FUNC IS 0
635 002026 001412          BEQ    1$           ;ALL SET, FINISH UP
636 002030 033715 001000'  BIT    BIT9,(R5)   ;IF BIT 9 OF XFER FUNCTION
637 002034 001407          BEQ    1$           ;ALL SET, FINISH UP
638 002036 012702 000276'  MOV    #BYTCC,R2   ;ELSE SET R2=ADDR OF BYTCC
639 002042 013737 000300' 000276'  MOV    WCC,BYTCC   ;MOVE VALUE OF WCC TO BYTCC AND
640 002050 006337 000276'  ASL    BYTCC       ;DOUBLE BY SHIFTING LEFT
641 002054          1$:
642 002054 063737 000314' 000340'  ADD    WBUF,WBFADR ;INC WBFADR ADDR BY VALUE IN WBUF
643 002062 012703 000340'  MOV    #WBFADR,R3 ;SET R3 = TO ADDR OF WBFADR
644 002066 000207          RTS    PC           ;RETURN
645
646
647
648
649
650
651 002070          RREGS:
652 002070 012702 000302'  MOV    #RCC,R2     ;SET R2 FOR A DATI CYCLE COUNT
653 002074 063737 000312' 000334'  ADD    RBUF,RBFADR ;INC RBFADR ADDR BY VALUE IN RBUF
654 002102 012703 000334'  MOV    #RBFADR,R3 ;SET R3 = TO VALUE IN RBFADR
655 002106 000207          RTS    PC           ;RETURN
656
657
658
659
660
661
662
663
664 002110          DIDATA:
665 002110 013703 000334'  MOV    RBFADR,R3   ;SET R3 = 1ST READ BUFF ADDR
666 002114 063703 000312'  ADD    RBUF,R3     ;GET LAST ADDR BY ADDING BUFF SIZE
667 002120 013704 000334'  MOV    RBFADR,R4   ;SET R4=PRESENT READ BUFF ADDR
668 002124          1$:
669 002124 021024          CMP    (R0),(R4)    ;IS VALLE IN DEV DATA REG=TO THAT IN READ BUFF?
670 002126 001407          BEQ    2$           ;IF EQUAL, CONTINUE
671 002130 104403 000000' 003174'  MSGN$,BEGIN,DIERR ;ASCII MESSAGE CALL WITH COMMON HEADER
672 002136 005744          TST    (R4)        ;RETURN R4 TO ERR LOCATION
673 002140 004737 002640'  JSR    PC,BDDATA   ;AND T'PE OUT CONTENTS OF REGS
674 002144 000402          BR    3$           ;GO EXIT
675 002146          2$:
676 002146 020403          CMP    R4,R3       ;IS R4 = LAST XFER LOCATION?
677 002150 103765          BLO   1$           ;IF NOT,CHECK NEXT ADDR
678 002152          3$:
679 002152 063737 000312' 000334'  ADD    RBUF,RBFADR ;INC READ BUFF ADDR BY RBUF VALUE
680 002160 000207          RTS    PC           ;RETURN
681
682
683

```

```

684
685 ;*****
686 ;THIS ROUTINE CHECKS FOR CORRECT DATO TRANSFERS ALL VALUES IN THE
687 ;WRITE TRANSFER BUFFER IS COMPARED WITH THE CONTENTS IN THE DEVICE
688 ;DATA BUFFER REGISTER
689 ;*****
690
691 DODATA:
692 002162 013703 000340' MOV WBFADR,R3 ;SET R3 = PRESENT WRITE BUFF ADDR
693 002166 063703 000314' ADD WBUF,R3 ;INC R3 BY # OF XFER LOCATIONS
694 002172 013704 000340' MOV WBFADR,R4 ;SET R4 = PRESENT WRITE BUFF ADDR
695 002176
696 002176 021024 1$: CMP (R0),(R4) ;IS DEV DATA REG = VALUE IN BUFF ADDR?
697 002200 001407 BEQ 2$ ;IF EQUAL, CONTINUE
698 002202 104403 000000' 003200' MSGN$,BEGIN,DOERR ;ASCII MESSAGE CALL WITH COMMON HEADER
699 002210 005744 TST -(R4) ;RETURN R4 TO ERR LOCATION
700 002212 004737 002640' JSR PC,BDDATA
701 002216 000402 BR 3$
702 002220
703 002220 020403 2$: CMP R4,R3 ;IS R4 AT LAST XFER LOCATION?
704 002222 103765 BLO 1$ ;IF NOT,GO CHECK NEXT ADDR
705 002224
706 002224 063737 000314' 000340' 3$: ADD WBUF,WBFADR ;INC WRITE BUFF BY VALUE IN WBUF
707 002232 000207 RTS PC ;RETURN
708 ;*****
709
710 ;*****
711 ;IF DATO (BITO EQ 1) SET EA BITS IN WBUFEA ;RS001
712 ;IF DATI (BITO EQ 0) SET EA BITS IN RBUFEA ;RS001
713 ;*****
714
715 EABITS: ;RS001
716 002234 005037 000310' CLR EABIT ;CLEAR EXTENDED MEMORY BITS ;RS001
717 002240 032737 000001 000266' BIT #BITO,DOEI ;DATO? ;RS001
718 002246 001003 BNE 1$ ;YES, SET EA BITS IN WBUFEA ;RS001
719 002250 004737 002264' JSR PC,REABITS ;NO, SET EA BITS IN RBUFEA ;RS001
720 002254 000402 BR 2$ ;RETURN ;RS001
721 002256 004737 002322' 1$: JSR PC,WEABITS ;SET EA BITS IN WBUFEA ;RS001
722 002262 000207 2$: RTS PC ;RETURN ;RS001
723
724 ;*****
725 ;THIS ROUTINE WILL SET THE CORRECT EXTENDED MEMORY ADDRESS BITS
726 ;OF THE CR2 REGISTER IN ACCORDANCE WITH THE EXTENDED MEMORY ADDRESS
727 ;BITS AS SET BY THE MONITOR IN RBUFEA.
728 ;*****
729
730 REABITS:
731 002264 032737 000020 000330' BIT #BIT4,RBUFEA ;IS BIT 4 OF RBUFEA SET?
732 002272 001403 BEQ 1$ ;IF NOT, CONTINUE
733 002274 052737 000001 000310' BIS #BIT0,EABIT ;ELSE SET BIT 0 OF EABIT
734 002302 032737 000040 000330' 1$: BIT #BIT5,RBUFEA ;IS BIT 5 OF RBUFEA SET?
735 002310 001403 BEQ 2$ ;IF NOT, RETURN
736 002312 052737 000002 000310' BIS #BIT1,EABIT ;ELSE SET BIT 1 OF EABIT

```



```

737 002320 000207      2$:   RTS      PC          ;RETURN
738
739
740                    ;*****
741                    ;THIS ROUTINE WILL PERFORM THE SAME FUNCTION AS THE REABITS ;RS001
742                    ;ROUTINE EXCEPT TO THE WBUFEA. ;RS001
743                    ;*****
744 002322      WEABITS:
745 002322 032737 000020 000322'   BIT      @BIT4,WBUFEA   ;IS BIT 4 OF WBUFEA SET? ;RS001
746 002330 001403      BEQ      1$          ;IN NOT, CONTINUE ;RS001
747 002332 052737 000001 000310'   BIS      @BIT0,EABIT   ;ELSE SET BIT 0 OF EABIT? ;RS001
748 002340 032737 000040 000322'  1$:   BIT      @BIT5,WBUFEA ;IS BIT 5 OF WBUFEA SET? ;RS001
749 002346 001403      BEQ      2$          ;IF NOT, RETURN ;RS001
750 002350 052737 000002 000310'   BIS      @BIT1,EABIT   ;ELSE SET BIT 1 OF EABIT ;RS001
751 002356 000207      2$:   RTS      PC          ;RETURN ;RS001
752
753
754                    ;*****
755
756                    ;THIS ROUTINE WILL CAUSE THE REQUEST LEVEL OF ALL DEVICES TO
757                    ;BE AT A LOWER LEVEL THAN IN THE PREVIOUS ITERATION---ONLY
758                    ;ONE LEVEL IS SET AT ANY ONE TIME EXCEPT FOR AN NPR ,THEN AN INTERRUPT
759                    ;LEVEL MUST BE SET FOR HARDWARE CONSIDERATIONS
760
761 002360      ROTRQS:
762 002360 022737 000060 000354'   CMP      @60,RQLVL     ;IS REQUEST AT NPR WITH BR7 INTR?
763 002366 001004      BNE      1$          ;IF NOT, CONTINUE
764 002370 012737 000020 000354'   MOV      @20,RQLVL     ;ELSE SET REQUEST LEVEL TO BR7
765 002376 000411      BR       2$          ;AND RETURN
766 002400      1$:
767 002400 006237 000354'   ASR      RQLVL         ;SHIFT REQUEST LEVEL TO NEXT BR
768 002404 022737 000001 000354'   CMP      @1,RQLVL     ;IF REQUEST LEVEL DOES NOT = 1
769 002412 001003      BNE      2$          ;THEN RETURN
770 002414 012737 000060 000354'   MOV      @60,RQLVL     ;ELSE RESET REQUEST TO NPR/BR7 INTR
771 002422
772 002422 000207      2$:   RTS      PC          ;RETURN
773
774                    ;*****
775                    ;*****
776                    ;THIS ROUTINE WILL ROTATE THE DATA PATTERNS TO TRY TO EXERCISE
777                    ;THE DATA LINES.
778                    ;*****
779                    ;*****
780 002424      ROTPAT:
781 002424 012701 000226'   MOV      @P2,R1        ;INIT R1 TO ADDR OF P2
782 002430      1$:
783 002430 032711 000001      BIT      @BIT0,(R1)     ;IF BIT 0 OF PATTERN IS 0
784 002434 001402      BEQ      2$          ;GO CLEAR C-BIT
785 002436 000261      SEC
786 002440 000401      BR       3$          ;AND CONTINUE
787 002442      2$:
788 002442 000241      CLC
789 002444      3$:
790 002444 006011      ROR      (R1)         ;ROTATE DATA TO THE RIGHT
791 002446 022701 000232'   CMP      @P4,R1        ;DOES R1 POINT TO LAST PATTERN?

```

```

792 002452 001403
793 002454 012701 000232'
794 002460 000763
795 002462
796 002462 000207
797
798
799
800
801
802
803
804 002464
805 002464 012702 004406'
806
807 002470 013703 000226'
808 002474 012704 004406'
809 002500 063702 000312'
810 002504
811 002504 010324
812 002506 022704 005344'
813 002512 103415
814 002514 020204
815 002516 101372
816 002520 063702 000312'
817 002524 023703 000232'
818 002530 001403
819 002532 013703 000232'
820 002536 000762
821 002540
822 002540 013703 000226'
823 002544 000757
824 002546
825 002546 012704 003444'
826 002552
827 002552 005024
828 002554 022704 004402'
829 002560 101374
830 002562 000207
831
832
833
834
835
836
837 002564
838
839 002564 012701 000001
840 002570 013700 000006'
841 002574 005004
842 002576
843 002576 030137 000240'
844 002602 001007
845 002604

```

```

      BEQ      4$          ;IF IT DOES,EXIT
      MOV      @P4,R1     ;ELSE SET R1=LAST PATTERN ADDR
      BR       1$          ;AND GO BACK FOR MORE
4$:
      RTS      PC          ;RETURN
;*****
;THIS ROUTINE WILL SET UP THE READ BUFFER WITH ONE OR BOTH DATA
;PATTERNS DEPENDING ON THE NUMBER OF DEVICES AND CLEAR THE WRITE
;BUFFER SO THE FOLLOWING TRANSFERS CAN BE COMPARED CORRECTLY.
;*****
RESDAT:
      MOV      @REEDIN,R2  ;SET R2=1ST ADDR OF READ BUFF
      MOV      P2,R3      ;SET R3= VALUE IN P2
      MOV      @REEDIN,R4 ;SET R4= ADDR OF READ BUFFER
      ADD      RBUF,R2    ;INC R2 TO LAST XFER ADDR
1$:
      MOV      R3,(R4)+   ;MOVE PATTERN INTO READ BUFF ADDR
      CMP      @REEDIN+736,R4 ;IS R4 LAST ADDR IN READ BUFF?
      BLO      3$          ;IF IT IS,GO TAKE CARE OF IT
      CMP      R2,R4      ;IS R4 AT LAST XFER ADDR?
      BHI      1$          ;IF NOT,GO TO NEXT ADDR
      ADD      RBUF,R2    ;INC R2 TO LAST ADDR OF NEXT XFER
      CMP      P4,R3      ;IS R3 = VALUE IN P4?
      BEQ      2$          ;IF IT IS,GO TAKE CARE OF IT
      MOV      P4,R3      ;ELSE SET R3 = P4
      BR       1$          ;AND GO AGAIN
2$:
      MOV      P2,R3      ;RESET R3 = P2
      BR       1$          ;AND GO AGAIN
3$:
      MOV      @WRTOUT,R4 ;SET R4 TO WRITE BUFFER
4$:
      CLR      (R4)+      ;CLEAR BUFFER
      CMP      @WRTOUT+736,R4 ;END OF BUFFER?
      BHI      4$          ;IF NOT,CLEAR NEXT WORD OF BUFF
      RTS      PC          ;RETURN
;*****
;*****
;THIS ROUTINE WILL COMPARE BITS IN DV WITH THOSE IN MASK TO
;DETERMINE WHICH DEVICES DID NOT INTERRUPT, THEN PRINT THEM.
;*****
GETDEV:
      MOV      @1,R1      ;SET BIT 0 IN R1
      MOV      ADDR,R0    ;INIT R0 TO 1ST DEV ADDR
      CLR      R4         ;CLEAR R4 AS NOT USED IN BDDATA RTN
1$:
      BIT      R1,DV      ;BIT TEST DV TO SEE IF DEV THERE
      BNE      3$          ;IF THERE, CHECK MASK REG.
2$:

```

```

846 002604 006301          ASL      R1           ;ELSE SHIFT TO NEXT BIT
847 002606 032701 010000    BIT      &10000,R1     ;AT BIT 12?
848 002612 001011          BNE     GETOUT        ;IF YES, EXIT
849 002614 062700 000020    ADD     &20,R0        ;ELSE GET NEXT DEV ADDR
850 002620 000766          BR      1$           ;AND SEE IF ITS THERE
851 002622          3$:
852 002622 030137 000242'    BIT      R1,MASK      ;DID DEV INTR?
853 002626 001366          BNE     2$           ;IF YES, GO CHECK NEXT ONF
854 002630 004737 002640'    JSR     PC,BDDATA     ;ELSE REPORT ALL ITS REGS.
855 002634 000763          BR      2$           ;THEN GO CHECK NEXT BIT
856 002636          GETOUT:
857 002636 000207          RTS      PC           ;EXIT
858          ;*****
859          ;*****
860          ;*****
861          ;THIS ROUTINE WILL LIST ALL THE CONTENTS OF THE DEVICE REGISTERS
862          ;ON REGISTERING AN INCORRECT DATA COMPARISON OR FOR AN INTERRUPT
863          ;ON ERROR
864          ;*****
865 002640          BDDATA:
866 002640 010437 000374'    MOV     R4,BADR       ;SET UP BAD MEM FOR ERR TYPEDUT
867 002644 010037 000360'    MOV     R0,DB         ;DB HAS DEV DATA REG ADDR
868 002650 013737 000360' 000362'    MOV     DB,CC         ;MOVE DB TO CC AND ADD 2
869 002656 062737 000002 000362'    ADD     &2,CC         ;TO GET DEV CYCLE COUNT REG ADDR
870 002664 013737 000362' 000364'    MOV     CC,BA         ;MOVE CC TO BA AND ADD 2
871 002672 062737 000002 000364'    ADD     &2,BA         ;TO GET DEV BUFF ADDR REG ADDR
872 002700 013737 000364' 000366'    MOV     BA,CR1        ;MOVE BA TO CR1 AND ADD 2
873 002706 062737 000002 000366'    ADD     &2,CR1        ;TO GET DEV CR1 REG ADDR
874 002714 013737 000366' 000370'    MOV     CR1,CR2       ;MOVE CR1 TO CR2 AND ADD 10
875 002722 062737 000010 000370'    ADD     &10,CR2       ;TO GET DEV CR2 REG ADDR
876 002730 013737 000366' 000100'    MOV     CR1,CSRA      ;GET BECR1 ADDR
877 002736 017737 175424 000102'    MOV     &BCR1,ACSR    ;GET CONTENTS OF BECR1
878
879 002744 012737 000001 000106'    MOV     &1,ERRTYP     ;DATA ERROR
880          ;*****
881          002752 104405 000000' 000360'
882          ;*****
883          CLR      10(R0)      ;CLEAR ALL ERRORS
884          RTS      PC           ;RETURN
885          ;*****
886          ;*****
887          ;THIS ROUTINE WILL SWAP READ CYCLE COUNT AND BUFFER SIZE WITH THE
888          ;WRITE CYCLE COUNT AND BUFFER SIZE ON AN ODD NUMBER OF DEVICES
889          ;BECAUSE ON THE FIRST ITERATION, ASSUMING A DEVICE COUNT OF 5,
890          ;3 OF THOSE DEVICES WILL BE DOING DATI(S) AND THE BUFFER SIZE
891          ;WILL BE SMALLER THAN THE 2 DEVICES DOING DATO(S). THEN ON THE
892          ;NEXT ITERATION THERE WILL BE 2 DEVICES DOING DATI(S) AND
893          ;3 DOING DATO(S) SO THE CYCLE COUNT AND BUFFER SIZES MUST REFLECT
894          ;THIS CHANGE.
895          ;*****
896 002766          SWAPCC:

```

```

897 002766 013702 000302'      MOV      RCC,R2      ;SAVE RCC
898 002772 013737 000300' 000302'  MOV      WCC,RCC    ;GET NEW RCC FROM WCC
899 003000 010237 000300'      MOV      R2,WCC     ;GET NEW WCC FROM OLD RCC
900 003004 013702 000312'      MOV      RBUF,R2    ;SAVE RBUF
901 003010 013737 000314' 000312'  MOV      WBUF,RBUF  ;GET NEW RBUF FROM WBUF
902 003016 010237 000314'      MOV      R2,WBUF    ;GET NEW WBUF FROM OLD RBUF
903 003022 000207      RTS      PC         ;RETURN
904
905
906 003024      ISR1:
003024 052737 000001 000242'  BIS      @BIT0,MASK ;SET BIT0 TO INDICATE DEV 1 INTERR'D
003032 000002      RTI      ;RETURN
907 003034      ISR2:
003034 052737 000002 000242'  BIS      @BIT1,MASK ;SET BIT1 TO INDICATE DEV 2 INTERR'D
003042 000002      RTI      ;RETURN
908 003044      ISR3:
003044 052737 000004 000242'  BIS      @BIT2,MASK ;SET BIT2 TO INDICATE DEV 3 INTERR'D
003052 000002      RTI      ;RETURN
909 003054      ISR4:
003054 052737 000010 000242'  BIS      @BIT3,MASK ;SET BIT3 TO INDICATE DEV 4 INTERR'D
003062 000002      RTI      ;RETURN
910 003064      ISR5:
003064 052737 000020 000242'  BIS      @BIT4,MASK ;SET BIT4 TO INDICATE DEV 5 INTERR'D
003072 000002      RTI      ;RETURN
911 003074      ISR6:
003074 052737 000040 000242'  BIS      @BIT5,MASK ;SET BIT5 TO INDICATE DEV 6 INTERR'D
003102 000002      RTI      ;RETURN
912 003104      ISR7:
003104 052737 000100 000242'  BIS      @BIT6,MASK ;SET BIT6 TO INDICATE DEV 7 INTERR'D
003112 000002      RTI      ;RETURN
913 003114      ISR8:
003114 052737 000200 000242'  BIS      @BIT7,MASK ;SET BIT7 TO INDICATE DEV 8 INTERR'D
003122 000002      RTI      ;RETURN
914 003124      ISR9:
003124 052737 000400 000242'  BIS      @BIT8,MASK ;SET BIT8 TO INDICATE DEV 9 INTERR'D
003132 000002      RTI      ;RETURN
915 003134      ISR10:
003134 052737 001000 000242'  BIS      @BIT9,MASK ;SET BIT9 TO INDICATE DEV 10 INTERR'D
003142 000002      RTI      ;RETURN
916 003144      ISR11:
003144 052737 002000 000242'  BIS      @BIT10,MASK ;SET BIT10 TO INDICATE DEV 11 INTERR'D
003152 000002      RTI      ;RETURN
917 003154      ISR12:
003154 052737 004000 000242'  BIS      @BIT11,MASK ;SET BIT11 TO INDICATE DEV 12 INTERR'D
003162 000002      RTI      ;RETURN
918
919 003164      NOSET: MMSG3
920 003166      177777
921 003170      ERRSET: MMSG4
922 003172      177777
923 003174      DIERR: MMSG1
924 003176      177777
925 003200      DOERR: MMSG2
926 003202      177777
927 003204      NOINTR: MMSG5
928 003206      177777
929 003210      045      104      101 MMSG1: .ASCIZ 'DATI OR DATIP ERROR'

```

	003213	124	111	040	
	003216	117	122	040	
	003221	104	101	124	
	003224	111	120	040	
	003227	105	122	122	
	003232	117	122	000	
930	003235	045	104	101	MSSG2: .ASCIZ ' *DATO OR DATOB ERROR'
	003240	124	117	040	
	003243	117	122	040	
	003246	104	101	124	
	003251	117	102	040	
	003254	105	122	122	
	003257	117	122	000	
931	003262	045	104	105	MSSG3: .ASCIZ ' *DEVICE READY BIT NOT SET'
	003265	126	111	103	
	003270	105	040	122	
	003273	105	101	104	
	003276	131	040	102	
	003301	111	124	040	
	003304	116	117	124	
	003307	040	123	105	
	003312	124	000		
932	003314	045	104	105	MSSG4: .ASCIZ ' *DEVICE INTERRUPTED ON ERROR--NOT ON DONE'
	003317	126	111	103	
	003322	105	040	111	
	003325	116	124	105	
	003330	122	122	125	
	003333	120	124	105	
	003336	104	040	117	
	003341	116	040	105	
	003344	122	122	117	
	003347	122	055	055	
	003352	116	117	124	
	003355	040	117	116	
	003360	040	104	117	
933	003363	116	105	000	
	003366	045	124	110	MSSG5: .ASCIZ ' *THE FOLLOWING DEVICE(S) DID NOT INTERRUPT; *'
	003371	105	040	106	
	003374	117	114	114	
	003377	117	127	111	
	003402	116	107	040	
	003405	104	105	126	
	003410	111	103	105	
	003413	050	123	051	
	003416	040	104	111	
	003421	104	040	116	
	003424	117	124	040	
	003427	111	116	124	
	003432	105	122	122	
	003435	125	120	124	
	003440	072	045	000	
934					. EVEN
935	003444				WRTOU: .BLKW 361
936	004406				REEDIN: .BLKW 361
937					
938	005350'				. "
939					

J2

BEAC DEC/X11 SYSTEM EXERCISER M MACRO M1200 30-MAY-84 09:15 PAGE 8-14
DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

SEQ 22

940

000001

.END

ACSR	000102R	CR2	000370R	ISR1	003024R	PRHMS#	000002	SAVR1	000272R
ADDR	000006R	CSRA	000100R	ISR10	003134R	PRTY	000000	SAVR5	000274R
ADDR22#	001000	DATAACK	001132R	ISR11	003144R	PRTY0	000000	SBADR	000102R
APTPRE#	000200	DATACK#	104411	ISR12	003154R	PRTY1	000040	SETUP	001410R
ASB	000106R	DATER#	104404	ISR2	003034R	PRTY2	000100	SIMLGO	000236R
ASTAT	000104R	DB	000360R	ISR3	003044R	PRTY3	000140	SOF CNT	000042R
AUTO	000010	DEV CNT	000264R	ISR4	003054R	PRTY4	000200	SOFER#	104405
AWAS	000110R	DICNT	000306R	ISR5	003064R	PRTY5	000240	SOF PAS	000045R
BA	000364R	DIDATA	002110R	ISR6	003074R	PRTY6	000300	SPOINT	000032R
BADMEM	000372R	DIERR	003174R	ISR7	003104R	PRTY7	000340	SPSIZ	000040
BADR	000374R	DOCNT	000304R	ISR8	003114R	PS	177776	SR1	000016R
BDDATA	002640R	DODATA	002162R	ISR9	003124R	PSW	177776	SR2	000020R
BEGIN	000000R	DODI	000266R	KTPRES#	000400	PTR1	000400R	SR3	000022R
BIT0	000001	DOERR	003200R	KTXTND#	040000	PTR2	000402R	SR4	000024R
BIT1	000002	DROP	001404R	MAP22#	104416	PUSH	005746	START	000410R
BIT10	002000	DV	000240R	MASK	000242R	PUSH2	024646	STAT	000026R
BIT11	004000	DVID1	000014R	MODNAM	000000R	PWRFLG#	000002	SVDODI	000270R
BIT12	010000	DVMASK	000400R	MODSP	000224R	P1	000224R	SVR0	000062R
BIT13	020000	DVREGS	000360R	MORDEV	001170R	P2	000226R	SVR1	000064R
BIT14	040000	EABIT	000310R	MORE	000244R	P3	000230R	SVR2	000066R
BIT15	100000	EABITS	002234R	MSGN#	104403	P4	000232R	SVR3	000070R
BIT2	000004	ECCMEM#	000100	MSG#	104402	QM(N)22#	000010	SVR4	000072R
BIT3	000010	ENDIT#	104413	MSSG1	003210R	RAND#	104417	SVR5	000074R
BIT4	000020	ENDRBF	000336R	MSSG2	003235R	RANNUM	000054R	SVR6	000076R
BIT5	000040	ENDWBF	000342R	MSSG3	003262R	REFADR	000334R	SWAPCC	002766R
BIT6	000100	END#	104410	MSSG4	003314R	RBUF	000312R	SYSCNT	000052R
BIT7	000200	ERBITS	000356R	MSSG5	003366R	RBUFEA	000330R	TOCNT	000405R
BIT8	000400	ERRSET	003170R	NCPUOP#	000020	RBUFPA	000326R	TRPDFD#	000025
BIT9	001000	ERRTYP	000106R	NOAPTY#	000002	RBUFSZ	000332R	USTACK#	000000
BREAK#	104407	EXINC	001644R	NOINTR	003204R	RBUFVA	000324R	VECTOR	000000
BRK	001070R	EXIT#	104400	NOSET	003164R	RCC	000302R	WASADR	000000
BR1	000012R	GETBUF	001520R	NPASS	001320R	REABIT	002264R	WBFADR	000000
BR2	000013R	GETDEV	002564R	NULL	000000	REEDIN	004406R	WBUF	000000
BTOD#	104421	GETOUT	002636R	NXT	001310R	REREST	000634R	WBUFEA	000328R
BUFSZ	000250R	GETPA#	104415	OPEN	000000	RESDAT	002464R	WBUFPA	000320R
BYTCC	000276R	GO	001054R	OTOA#	104420	RESTR	000534R	WBUFVA	000316R
B12T15	000246R	GWBUF#	104414	PARPRE#	002000	RES1	000056R	WCC	000300R
CAPRES#	000004	HRDCNT	000044R	PASCNT	000034R	RES2	000060R	WDFR	000116R
CC	000352R	HRDER#	104405	PDPF11#	000002	RH70	001000	WDT0	000114R
CDATA#	104412	HRDPAS	000050R	PDP44	100000	ROTCNT	000234R	WEABIT	002352R
CHEKPX	001742R	ICONT	000036R	PDP60	001000	ROTPAT	002424R	WREGS	002016R
CKHNG#	000001	ICOUNT	000040R	PDP70	010000	ROTRQS	002360R	WRTOUT	003040R
CKMORE	001436R	IDNUM	000122R	PIRQ#	000004	RQLVL	000354R	XFLAG	000000R
CLKPRE#	000001	INCCNT	001622R	POPSP	005726	RREGS	002070R	XFR1	000344R
CLKSP#	104422	INCDO	001640R	POPSP2#	022626	RSTRT	000112R	XFR2	000346R
CLRREG	001646R	INDPAR#	000040			R6	0000006	XFR3	000350R
CONFIG	000056R	INIT	000030R			R7	0000007	XFR4	000352R
CR1	000366R	INTR	000120R						

. ABS. 000000 000
005350 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 13417 WORDS (53 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:01:02
BEA, BEA/ - SP=DDXCOM, BEA

DEC/X11 SYSTEM EXERS....B1
DEC/X11 SYSTEM EXERS....C1
DEC/X11 SYSTEM EXERS....D1
DEC/X11 SYSTEM EXERS....E1
DEC/X11 SYSTEM EXERS....F1
DEC/X11 SYSTEM EXERS....G1
DEC/X11 SYSTEM EXERS....H1
DEC/X11 SYSTEM EXERS....I1
DEC/X11 SYSTEM EXERS....J1
DEC/X11 SYSTEM EXERS....K1
DEC/X11 SYSTEM EXERS....L1
DEC/X11 SYSTEM EXERS....M1
DEC/X11 SYSTEM EXERS....N1

DEC/X11 SYSTEM EXERS....B2
DEC/X11 SYSTEM EXERS....C2
DEC/X11 SYSTEM EXERS....D2
DEC/X11 SYSTEM EXERS....E2
DEC/X11 SYSTEM EXERS....F2
DEC/X11 SYSTEM EXERS....G2
DEC/X11 SYSTEM EXERS....H2
DEC/X11 SYSTEM EXERS....I2
DEC/X11 SYSTEM EXERS....J2
SYMBOL TABLEK2