

.REY I

IDENTIFICATION

PRODUCT CODE: AC-F072B-MC
PRODUCT NAME: CXBMH80 M9312 MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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

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

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

COPYRIGHT (C) 1978 DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

THIS MODULE PERFORMS CHECKSUM VERIFICATION OF THE M9312 BOOTSTRAP TERMINATOR. IT COMPARES THE CHECKSUM FOUND IN THE LAST LOCATION OF THE ROMS TO ONE IT CALCULATES FROM ALL OTHER LOCATIONS. OTHER THAN THE EXCEPTION LOCATION, LOCATION SRI IS USED TO SELECT WHICH ROMS TO TEST.

2.0 REQUIREMENTS

HARDWARE: ANY PDP-11 PROCESSOR WITH A M9312 BOOTSTRAP TERMINATOR AND AT LEAST ONE ROM PLUGGED INTO THE TERMINATOR.

STORAGE: BMH REQUIRES:

1. DECIMAL WORDS: 237
2. OCTAL WORDS: 0355
3. OCTAL BYTES: 732

3.0 PASS DEFINITION

ONE PASS CONSISTS OF DOING A CHECKSUM ON EACH ROM 30 (8) TIMES.

4.0 EXECUTION TIME

BMH TAKES APPROXIMATELY 35 SECONDS TO COMPLETE A PASS WHEN RUNNING ALONE.

5.0 CONFIGURATION REQUIREMENTS

SET THE CORRESPONDING BITS IN SRI TO A"1" FOR THE DESIRED ROMS:

BIT 0 = 1	-	DIAG. ROM	IN	E-35	{173000-173177}
BIT 1 = 1	-	BOOT ROM	IN	E-33	{173200-173377}
BIT 2 = 1	-	BOOT ROM	IN	E-34	{173400-173577}
BIT 3 = 1	-	BOOT ROM	IN	E-32	{173600-173777}

6.0 DEVICE/OPTION SETUP

BMHB DEC/X11 SYSTEM EXERCISER MODULE
XBMHBO.P11 12-OCT-78 11:52

MACY11 30A(1052) 12-OCT-78 16:22 PAGE 4

SEQ 0003

NONE

7.0 MODULE OPERATION

THIS MODULE FIRST CHECKS FOR SR1 RO BE NON-ZERO. IF IT IS ZERO THE MODULE TYPES THE MESSAGE:

NO ROMS SELECTED SR 1 = 0

AND THEN NOTIFIES THE MONITOR TO DROP IT. IF SR1 IS NON-ZERO THE CONTROL ROUTINE THEN CALLS THE "CEPROM" SUBROUTINE TO LOCATE THE FIRST ROM IF THE LOCATION IS NOT SELECTED. IF THE "CEPROM" SUBROUTINE RETURNS WITH "FIRSTA" = 0, THE "CEPROM" WILL RETURN WITH "FIRSTA" CONTAINING THE FIRST ADDRESS TO BE STASHED. CONTAINING THE LAST ADDRESS TO BE SUMMED AND LOCATION "LASTA" CONTAINING THE ADDRESS OF THE ROMS CHECKSUM.

THE CONTROL ROUTINE THEN CALLS THE SUB-ROUTINE "CHECKR" TO CHECK THE ROM IF LOCATION "FIRSTA" EQUALS 780. "CHECKR" WILL JUST RETURN TO THE CONTROL ROUTINE. BUT "CHECKR" WILL CALL THE "CEPROM" SUBROUTINE TO LOCATE THE FIRST ROM IF "FIRSTA" IS NOT SELECTED. "CHECKR" WILL RETURN WITH "FIRSTA" = 0, WITH THE EXCEPTION OF "EXCADR" OF THE ROM FROM WHICH THE CALCULATED CHECKSUM IN LOCATION "BAD" IS THE ROUTINE "CHECKR" WITH COMPARES THE GOOD CHECKSUM FROM THE BOARD WITH THE "BAD" SUM AND IF THEY DON'T COMPARE REPORTS AN ERROR.

CHECKSUM ERROR ON M9312 BOOTSTRAP.

THE ROUTINE "CHECKR" THEN RETURNS TO THE CONTROL ROUTINE AND THE PROCESS IS REPEATED FOR EACH ROM.

8.0 OPERATING OPTIONS

NONE

9.0 NON-STANDARD PRINTOUTS

NONE

```

000000* BKMOD <BMHB > 0,0,0,0,0,30,164
000000* 40020, BMHB 0,0,0,0,0,30,164
; TITLE BMHB DEC/X11 SYSTEM EXERCISER MODULE
DOXCOM VPRSION 6 23-MAY-78
*****LIST BIN*****
000000* BEGIN:
000000* 046502 041110 040 MODNAM: .ASCII /BMHB / ;MODULE NAME.
000005* 000 ADDR: .BYTE OPEN ;USED TO KEEP TRACK OF WBOFF USAGE
000006* 000000 ADDR: 0+0 ;1ST DEVICE ADDR
000010* 000000 VECTOR: 0+0 ;1ST DEVICE VECTOR.
000012* 000 BR1: .BYTE PRTVO+0 ;1ST RR LEVEL.
000013* 000 BR2: .BYTE PRTVO+0 ;2ND RR LEVEL.
000014* 000001 DVID1: 0+1 ;DEVICE INDICATOR 1.
000016* 000000 SR1: OPEN ;SWITCH REGISTER 1.
000020* 000000 SR2: OPEN ;SWITCH REGISTER 2.
000022* 000000 SR3: OPEN ;SWITCH REGISTER 3.
000024* 000000 SR4: OPEN ;SWITCH REGISTER 4.
*****
000026* 040020 STAT: 40020 ;STATUS WORD.
000030* 000252 INIT: START ;MODULE START ADDR.
000032* 000224 SPOINT: MODSP ;MODULE STACK POINTER.
000034* 000000 PASCNT: 0 ;PASS COUNTER.
000036* 000030 ICOUNT: 30 ;# OF ITERATIONS PER PASS=30
000040* 000000 ICONF: 0 ;LOC TO COUNT ITERATIONS
000042* 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044* 000000 HRDCT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046* 000000 SOPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050* 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052* 000000 SVSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054* 000000 RANMUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056* 000000 CONFIG: 0 ;RESERVED FOR MONITOR USE
000056* 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000060* 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062* 000000 SVR0: OPEN ;LOC TO SAVE R0.
000064* 000000 SVR1: OPEN ;LOC TO SAVE R1.
000066* 000000 SVR2: OPEN ;LOC TO SAVE R2.
000070* 000000 SVR3: OPEN ;LOC TO SAVE R3.
000072* 000000 SVR4: OPEN ;LOC TO SAVE R4.
000074* 000000 SVR5: OPEN ;LOC TO SAVE R5.
000076* 000000 SVR6: OPEN ;LOC TO SAVE R6.
000100* 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
000102* 000000 SRADR: OPEN ;ADDR OF GOOD DATA, OR
000104* 000000 WASADR: OPEN ;ADDR OF BAD DATA, OR
000104* 000000 ASSTAT: OPEN ;STATUS REG CONTENTS.
000106* 000000 ERATYP: OPEN ;TYPE OF ERROR
000106* 000000 ASB: OPEN ;EXPECTED DATA.
000110* 000000 AWAS: OPEN ;ACTUAL DATA.
000112* 000252 RSTRT: RFSRTR ;RESTART ADDRPS AFTER END OF PASS
000114* 000000 WDTOT: OPEN ;WORDS TO MEMORY PER ITERATION
000116* 000000 WDFP: OPEN ;WORDS FROM MEMORY PER ITERATION
000120* 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000122* 000164 IDNUM: 164 ;MODULE IDENTIFICATION NUMBER=164
000040 000040 ;MODULE STACK STARTS HERE.
;LIST
;SPSTZ

```

```

;WORD 0
;LIST
;FNDR
000224*
*****
200
201 000224* 000000 FIRSTA: 0
202 000226* 000000 EXCADR: 0
203 000230* 000000 LAST: 0
204 000232* 000000 CROMP: 0
205 000234* 000000 GOODA: 0
206 000236* 000000 BAD: 0
207 000240* 000000 COUNT: 0
208 000242* 000236* BADADR: #BAD ;+ADDRESS POINTER FOR ERHORN CALL
209 000244* 000234* CRCER1: #GOODA ;+ADDRESS OF EXPECTED CHECKSUM
210 000246* 000232* #BADADR ;+ADDRESS OF ACTUAL CHECKSUM
211 000250* 174777 -1 ;+MESSAGE TERMINATOR

```

```

212 000252* 005767 177540 START: TST SRI ;ARE ANY ROMS TO BE CHECKED
213 000252* 005767 177540 RESTR: BNE ZS ;IF YES DO TEST
214 000252* 011095 MSGNS,BEGIN,MSG1 ;ASCII MESSAGE CALL WITH COMMON HEADER
215 000250* 104403 000000* 000622* 2S: MOV #1, CROMP ;INITIALIZE CURRENT ROM POINTER
216 000266* 104410 000000* 177732 3S: CMP #40, CROMP ;HAVE ALL ROMS BEEN TESTED
217 000272* 012767 000001 177732 BFC ;IF YES WE'RE DONE
218 000306* 027467 000040 177724 JSR PC, GETROM ;GET ROM ADDRESSES
219 000310* 004767 000014 JSR PC, CHECKR ;GO TEST ROM
220 000314* 004767 000146 BR 3S ;CONTINUE TESTING
221 000322* 104413 000000* 4S: ENDTTS,BEGIN ;SIGNAL END OF ITERATION.
222 000326* 000761 BR 2S ;MONITOR SHALL TEST END OF PASS
223 000330* 005067 177670 GETROM: CLR FIRSTA ;CLEAR FIRST CHECKSUM ADDRESS
224 000332* 036767 177672 177454 BIT CROMP, SRI ;IS CURRENT ROM TO BE TESTED
225 000344* 016700 177662 MOV 4S ;IF NO THEN JUST EXIT
226 000350* 000241 CLC ;GET CURRENT ROM POINTER TO WORK ON
227 000352* 006000 MOV RO ;CARRY BIT TO BE USED AS INDICATOR
228 000356* 012767 165000 177640 BVC 1S ;IF C-BIT NOT SET THEN BOOT ROM
229 000364* 005067 177636 CLR #165000,FIRSTA ;BEING TEST SO GO SET UP FOR ROOT ROM
230 000370* 012767 165775 177632 EXCADR ;NO DIAG ROM SET UP ITS FIRST ADDRESS
231 000370* 012767 165775 177632 MOV #165775, LASTA ;NO EXCEPTION ADDRESS IN DIAG.ROM
232 000404* 000425 173000 MOV #165776, GOODA ;SET UP LAST ADDRESS
233 000406* 012701 173000 1S: MOV #173000,R1 ;SET UP CHECKSUM ADDRESS OF DTAG. ROM
234 000414* 006000 2S: ROR RO ;DIAG ROM SET UP SO GO EXIT
235 000416* 062701 000200 BCS 3S ;GET FIRST ROOT ROM ADDRESS FOR WORKING
236 000422* 000773 ADD #200, R1 ;IF C-BIT SETS WE'VE FOUND OUR ROM
237 000424* 010167 177574 BR 2S ;SO GO SET UP FOR ADDRESS FOR IT
238 000424* 010167 177574 3S: MOV R1, FIRSTA ;IF C-BIT CLEAR UPDATE TO NEXT ROM
239 000434* 010167 177566 MOV #24, R1 ;SET UP FIRST ADDRESS
240 000440* 062701 000151 ADD R1, EXCADR ;CALCULATE EXCEPTION ADDRESS
241 000444* 010167 177560 MOV R1, LASTA ;AND STORE IT
242 000450* 062701 000001 ADD #51, R1 ;AND STORE IT
243 000454* 010167 177554 MOV R1, LASTA ;CALCULATE GOOD CHECKSUM ADDRESS
244 000460* 006167 177546 4S: ROL CROMP ;AND STORE IT
245 000464* 000207 RTS PC ;UPDATE CROMP TO NEXT ROM
246 ;RETURN
247
248 000466* 005767 177532 CHECKR: TST FIRSTA ;IS FIRST ADDRESS ZERO
249 000474* 001415 BFC 1S ;IF YES DONT TEST JUST EXIT
250 000474* 001415 JSR PC, CALSUM ;IF NOT GO CALCULATE CHECKSUM
251 000500* 027767 177530 177530 CMP #GOODA, BAD ;DO BOTH CHECKSUMS COMPARE
252 000506* 001410 BFC 1S ;IF YES EXIT
253 000510* 104403 000000* 000626* MSGNS,BEGIN,MSG2 ;ASCII MESSAGE CALL WITH COMMON HEADER
254 000516* 005067 177364 CLR ERRTRP ;IF YES EXIT
255 000522* 104405 000000* 000244* HDRFRS,BEGIN,CRCERR ;CHECKSUM ERROR
256 ;*****
257 ;*****

```

```

268 000530* 000207 1S: RTS PC ;RETURN
269
270
271 000532* 016700 177466 CALSUM: MOV FIRSTA, RO ;GET STARTING ADDRESS
272 000536* 005005 CLR R5 ;CLEAR CRC WORD
273 000540* 012003 LOOP: MOV (RO)+, R3 ;GET BYTE
274 000544* 012702 000020 CRCLOP: MOV #16,, R2 ;SET BIT COUNT
275 000546* 000241 CLC ;CLEAR CARRY
276 000550* 006005 ROR R5 ;ROTATE CRC WORD
277 000554* 006003 ROR R3 ;ROTATE BYTE
278 000554* 102006 BVC 1S ;ROTATE BYTE
279 000556* 012701 120001 MOV #120001,R1 ;GONE THROUGH HERE 16 TIMES
280 000562* 040501 120001 BIT #120001,R5 ;IF NO DO IT AGAIN
281 000564* 042705 120001 BIC #120001,R5 ;IS NEXT ADDR. AN EXCEPTION ADDR.
282 000570* 050105 1S: BIC R1, R5 ;NO CONTINUE
283 000574* 005302 R2 CRCLOP ;YES ADD 2 TO ADDR. TO SKIP IT
284 000574* 003364 BGT R2, EXCADR ;HAVE WE SUM THE WHOLE ROM
285 000576* 020067 177424 ROR R2, LASTA ;IF NO CONTINUE WITH NEXT BYTE
286 000600* 001001 TST (RO)+ ;IF YES STORE CALCULATED CHECKSUM
287 000606* 005720 2S: CMP R0, LASTA ;AND EXIT
288 000606* 020067 177416 BLOS LOOP ;IF YES STORE CALCULATED CHECKSUM
289 000612* 101752 MOV R5, BAD ;AND EXIT
290 000614* 010567 177416 JSR PC, BAD ;AND EXIT
291 000620* 000207 RTS PC ;AND EXIT
292
293
294 000622* 000632* MSG1: YORROMS
295 000624* 177777 MSG2: CRCERR
296 000626* 000664* 177777
297 000630* 177777
298
299 000632* 020045 047516 051040 NOROMS: .ASCIZ "% NO ROMS SELECTED SRI=0%"
300 000640* 046517 020123 042523
301 000646* 042514 052103 042105
302 000654* 051440 030522 030075
303 000662* 000045 CRCERR: .ASCIZ "% CHECKSUM ERROR ON M9312 ROOTSTRAP%"
304 000667* 051513 044103 041505
305 000675* 046525 042440
306 000700* 051127 051117 047440
307 000706* 020116 034515 030463
308 000722* 020062 047502 052117
309 000722* 052123 040522 022520
310 000730* 000
311 000732* 000732* .EVEN
312 000001 .END

```


SVR2	000066R	175#
SVR3	000070R	176#
SVR4	000072R	177#
SVR5	000074R	178#
SVR6	000076R	179#
SVSCNT	000052R	166#
TRPFD=	000022	200#
VECTDR	000010R	149#
WASADR	000104R	183#
WDPR	000116R	190#
WDFO	000114R	189#
XFLAG	00005R	147#
.	= 000732R	311#

. ABS. 000000 000
000732 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

XBH80, XBH80/SOL/CRF:SYM=DDXCON, XBH80
RUN-TIME: 1 1 2 SECONDS
RUN-TIME RATIO: 24/2=8.6
CORE USED: 7K (13 PAGES)