

IDENTIFICATION

PRODUCT CODE: AC-E9538-MC
PRODUCT NAME: CXDMC80 DMC-11 MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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

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

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

COPYRIGHT (C) 1976, 1978 DIGITAL EQUIPMENT CORPORATION

MAIN DEC CHANGE NOTICE
MAY BE REQUIRED FOR
PROGRAM TO OPERATE

1. ABSTRACT

DMC IS AN IOMOD THAT EXERCISES UP TO AND INCLUDING TWO CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED DMC11 SYNCHRONOUS INTERFACES. IT USES MAINTENANCE MODE OR A TURN AROUND CONNECTOR TO TRANSMIT AND RECEIVE A SET DATA PATTERN. THE RECEIVER AND TRANSMITTER ISR ARE ALWAYS AT PRIORITY FIVE (BR1/BR2). DATA CHECKING IS PERFORMED AT LEVEL 0 AND DONE OUTSIDE THE ISRS.

2. REQUIREMENTS

HARDWARE: AT LEAST 1 DMC11-AR WITH A
DMC11-DA OR DMC11-FA
OR
AT LEAST 1 DMC11-AL WITH A
DMC11-MA OR A DMC11-MD

STORAGE:: DMC REQUIRES:

1. DECIMAL WORDS: 1578
2. OCTAL WORDS: 03052
3. OCTAL BYTES: 6124

3. PASS DEFINITION

ONE PASS OF THE DMC MODULE CONSISTS OF TRANSMITTING AND RECEIVING 7 BUFFERS OF 100 CHARACTERS 200 TIMES FOR EACH SELECTED DEVICE.

4. EXECUTION TIME

RUNNING ALONE ON AN 11/45 ONE PASS TAKES APPROXIMATELY ONE MINUTE.

5. CONFIGURATION PARAMETERS.

DEFAULT PARAMETERS:
ADDR: 1 VECTOR: 1 BR1: 5 BR2: 5 DVID1: 1 AND SR1:0
DMC WILL RUN UP TO TWO CONSECUTIVELY ADDRESSED AND
CONSECUTIVELY VECTORED DMC11'S. IF SR1 IS ZERO THEN
LINE UNIT LOOP (MAINT MODE) IS USED AND UP TO 16
DMC11S (8 DMC MODULES) MAY BE RUN AT A TIME. IF SR1
BIT0 = 1 THEN LINE UNIT LOOP IS NOT USED AND A TURN-
AROUND TO RUN AT SPEED. IF YOU ARE RUNNING THE DMC11S AT
DMC TO RUN AT SPEED. IF YOU ARE RUNNING THE DMC11S AT
SPEED NO MORE THAN A TOTAL SPEED OF 2 MEGABAUD AT
FULL DUPLEX WILL RUN AT A TIME. THAT IS TO SAY IF
YOU ADD THE SPEEDS OF ALL DMC11S THAT ARE RUNNING
THE TOTAL MUST NOT EXCEED 2 MEGABAUD. EXAMPLES:

AT THE BAUD OF ONE MEG A MAXIMUM OF TWO DMC11S
CAN BE RUN AT A TIME.

AT THE BAUD OF 500K A MAXIMUM OF FOUR
DMC11S CAN BE RUN AT A TIME.

OR ANY COMBINATIN SUCH AS 1 DMC11 AT A MEG
AND 2 DMC11S AT 500K BAUD CAN BE RUN AT A TIME,
ANY COMBINATIN AS LONG AS THE TOTAL BAUD DOES
NOT EXCEED 2 MEG.

6. DEVICE/OPTION SETUP

IF SRI = 0 (LINE UNIT LOOP MODE) THEN THERE IS NO
SPECIAL SET UP NECESSARY.
IF SRI BIT0 = 1 (RUNNING THE DMC AT SPEED) THEN
A TURN-AROUND CONNECTOR MUST BE INSTALLED.

NOTE: SRI CAN BE SET UP AT CONFIGURATION TIME OR
AT RUN TIME WITH A MOD COMMAND.

7.

MODULE OPERATION

1. LOAD SOFTWARE POINTERS IN LINK TABLE.
2. LOAD VECTORS AND PRIORITIES IN TABLE.
3. ENABLE SELECTED DEVICES.
4. SCAN FOR ALL DEVICES TO FINISH.
5. IF NOT DONE GO TO 4. AND DROP HUNG DEVICE.
6. CHECK DATA FOR ALL DEVICES SELECTED.
7. DECREMENT ITERATION COUNT
8. IF NOT = 0 GO TO 1
9. SIGNAL ENDPASS.

IISR: INPUT INTERRUPT SERVICE ROUTINE.

11. GET INTERRUPTING DMCSER.
12. IF BASE I WAS REQUESTED, LOAD BASE ADDRESS.
13. IF RECEIVE BA/CC WAS REQUESTED, LOAD REC BA/CC.
14. IF XMIT BA/CC WAS REQUESTED, LOAD XMIT BA/CC.
15. RTI

OISR: OUTPUT INTERRUPT SERVICE ROUTINE.

01. GET INTERRUPTING DMCSER
02. IF ERROR REPORT IT AND EXIT.
03. IF XMIT DONE OR REC DONE, SET APPROPRIATE BITS IN THE ENDPASS FLAG FOR THE DEVICE.
04. RTI

9. NON-STANDARD PRINTOUTS

IF THE MODULE "HANGS" IN WHICH NOT ALL SELECTED DEVICES HAVE FINISHED, THEN A "HUNG" MESSAGE IS PRINTED OUT. CHECK THE ENDPASS FLAGS FOR EACH SELECTED DEVICE IN THE LINK TABLE TO DETERMINE WHICH DEVICE FAILED TO FINISH AND HOW FAR IT GOT.

FOR EXAMPLE:

THE TWO ENDPASS FLAGS ARE LOCATED IN THE LINK TABLE (LINK) AT THE FOLLOWING LOCATIONS.

XX11:
XX21:

ONLY BITS 0 THRU 4 ARE USED AND ARE DEFINED AS FOLLOWS:
BIT0 = 1 THE BASE ADDRESS WAS LOADED.
BIT1 = 1 7 RECEIVE BA/CC'S WERE LOADED.
BIT2 = 1 7 TRANSMIT BA/CC'S WERE LOADED.
BIT3 = 1 7 RECEIVE DONE'S WERE RECEIVED.
BIT4 = 1 7 TRANSMIT DONE'S WERE RECEIVED.

A CORRECT END PASS FLAG = 37 WHEN THE ENDPASS FLAGS = 37 FOR THE SELECTED DEVICES. IF THE DATA IS CHECKED, IF A "HUNG" MESSAGE IS TYPED IT IS BECAUSE ONE OR BOTH DEVICES DID NOT FINISH TO FIND WHICH ONE, CHECK THE ENDPASS FLAGS. ANY THAT ARE NOT EQUAL TO 37 ARE THE HUNG DEVICES. CHECK WHICH BITS OF THE ENDPASS FLAG ARE CLEAR TO SEE WHAT IT WAS TRYING TO DO.

SOFT ERROR

IF THE DMC'S PROTOCOL CHECKERS DETECT AN ERROR IN THE TRANSMISSION OF A MESSAGE, IT WILL RE-TRANSMIT THE ENTIRE MESSAGE, UPDATING AN ERROR COUNTER IN IT'S RAM. IF THIS COUNTER EXCEEDS 7 ON ANY GIVEN MESSAGE, IT WILL DECLARE A HARD ERROR. HOWEVER, IF FEWER THAN 7 OCCUR, IT WILL TAKE NO NOTICE OF THE CONDITION. FOR DEC/XX11 PURPOSES, HOWEVER, THE DMC MODULE WILL CHECK THE ERROR COUNTER AFTER EACH MESSAGE. IF IT HAS BEEN INCREMENTED AT ALL, IF AT LEAST ONE RE-TRANSMISSION WAS MADE, DMC WILL DECLARE A SOFT ERROR.

THE SOFT ERROR MESSAGE MAY INDICATE AN INTERMITTANT DEVICE FAILURE OR OTHER HARDWARE PROBLEM; HOWEVER, IF THE MESSAGE OCCURS IN A HEAVILY LOADED SYSTEM, IT MAY BE THAT THE MESSAGE IS DUE TO BUS LATENCY. THE DMC DOES NOT RECOGNIZE A DISTINCT "DATA LATE" ERROR. IT CONSIDERS THE CONDITION MERELY ANOTHER TRANSMISSION PROBLEM. ESPECIALLY IF THERE ARE OTHER FAST DIRECT MEMORY ACCESS DEVICES. IT COULD BE THAT THE DMC-11'S NPR'S ARE NOT BEING HOWER QUICKLY ENOUGH TO PREVENT BIT-DROPPING. TO VERIFY WHETHER THIS IS THE CONDITION, RUN A SINGLE DMC MODULE WITH A SINGLE DMC-11 DEVICE SELECTED. IF THE MESSAGE SHOULD NOT OCCUR UNDER THESE CONDITIONS, THE SOFT ERROR IT DOES OCCUR, THE PROBLEM IS PROBABLY IN DMC-11 HARDWARE OR A CABLE FAULT.

DMCB DEC/X11 SYSTEM EXERCISER MODULE
XDMCBO-P11 12-OCT-78 12:02

MACY11 30A(1052) 12-OCT-78 16:29 PAGE 7

SEQ 0006

```
204 000000* IOMOD <DMCB >,1,1,5,5,0,200,127
205 000000* MODULE 140000,DMCB,1,1,5,5,0,200,127
206 ; TITLE DMCB DEC/X11 SYSTEM EXERCISER MODULE
207 ; DDXCDM VERSION 6 23-MAY-78
208 ; LIST BIN
209 *****
210 000000* BEGIN:
211 000000* 046504 041103 040 MODNAM: .ASCII /DMCB / ;MODULE NAME
212 000000* 000001 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
213 000000* 000001 ADDR: 1+0 ;1ST DEVICE ADDR.
214 000010* 000001 VECTOR: 1+0 ;1ST DEVICE VECTOR.
215 000014* 000001 BR1: .BYTE PRTV5+0 ;1ST BR LEVEL.
216 000014* 000001 BR2: .BYTE PRTV5+0 ;2ND BR LEVEL.
217 000016* 000000 DVID1: 0+1 ;DEVICE INDICATOR 1.
218 000020* 000000 SR1: OPEN ;SWITCH REGISTER 1
219 000020* 000000 SR2: OPEN ;SWITCH REGISTER 2
220 000024* 000000 SR3: OPEN ;SWITCH REGISTER 3
221 000024* 000000 SR4: OPEN ;SWITCH REGISTER 4
222 *****
223 000026* 140000 STAT: 140000 ;STATUS WORD
224 000030* 000256 INTT: START ;MODULE START ADDR.
225 000032* 000224 SPINT: MODSP ;MODULE STACK POINTER.
226 000034* 000000 PASCNT: 0 ;PASS COUNTER
227 000036* 000200 ICOUNT: 200 ;# OF ITERATIONS PER PASS=200
228 000040* 000000 SOFCNT: 0 ;LOC TO COUNT ITERATIONS
229 000042* 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
230 000044* 000000 SOFPAS: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
231 000046* 000000 HODPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
232 000052* 000000 SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
233 000054* 000000 RANNUM: 0 ;# OF SYS ERRORS ACCUMULATED
234 000056* 000000 CONFIG: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
235 000060* 000000 RES2: 0 ;RESERVED FOR MONITOR USE
236 000062* 000000 SVR0: OPEN ;RESERVED FOR MONITOR USE
237 000064* 000000 SVR1: OPEN ;LOC TO SAVE R0.
238 000066* 000000 SVR2: OPEN ;LOC TO SAVE R2.
239 000068* 000000 SVR3: OPEN ;LOC TO SAVE R3.
240 000070* 000000 SVR4: OPEN ;LOC TO SAVE R4.
241 000072* 000000 SVR5: OPEN ;LOC TO SAVE R5.
242 000074* 000000 SVR6: OPEN ;LOC TO SAVE R6.
243 000076* 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
244 000100* 000000 SBADR: OPEN ;ADDR OF GOOD DATA, OR
245 000102* 000000 ACSR: OPEN ;CONTENTS OF CSR.
246 000104* 000000 WABADR: OPEN ;ADDR OF BAD DATA, OR
247 000106* 000000 ASAT: OPEN ;STATUS REG CONTENTS.
248 000108* 000000 ERRTYP: 0 ;TYPE OF ERROR
249 000110* 000000 ASD: OPEN ;EXPECTED DATA.
250 000112* 000224 AMAS: OPEN ;ACTUAL DATA.
251 000114* 000000 RSTRT: RESTR ;RESTART ADDRESS AFTER END OF PASS
252 000116* 000000 WDR: OPEN ;WORDS TO MEMORY PER ITERATION
253 000120* 000000 INTR: OPEN ;WORDS FROM MEMORY PER ITERATION
254 000122* 000127 IDNUM: 127 ;# OF INTERRUPTS PER ITERATION
255 000224* 000127 MODSP: ;MODULE IDENTIFICATION NUMBER=127
256 *****
257 *****
258 *****
259 *****
```

```
260 ;VARIABLES FOR DMC11
261 DLY1: 0
262 DLY2: 0
263 SELECT: 0
264 FLAGB: 37
265 FIRST: 0
266 MASK: 0
267 RCOUNT: 100
268 TCOUNT: 100
269 VA: 0
270 PA: 0
271 EA: 0
272 SAR0: 0
273 SAR1: 0
274 SAR2: 0
275 SAR3: 0
276 TERM = 0
277
```

```
278 ; BEGIN THE TESTS FOR THE DMC11
279
280
281 000256 012767 000010 177634 START: MOV #8,INTR ;8 INTERRUPTS/ITERATION
282 000272 012767 000700 177622 MOV #700,WDTD ;700 WORDS TO MEM/ITERATION
283 000272 012767 000700 177622 MOV #700,WDR ;700 WORDS FROM MEM/ITERATION
284 000300 032767 177774 177506 BIT #C<3>,DVID1 ;DROP MODULE IF DEVICES OTHER
285 000300 001004 177500 177712 BNE DROP ;THAN FIRST 4 ARE SELECTED
286 000316 001002 MOV #DVID1,SELECT ;SELECT=ACTIVE DEVICES
287 000316 001002 BNE RESTR1 ;DROP MODULE IF NO ACTIVE DEVICES
288
289 000320 104410 000000 DROP: ENDS,BEGIN ;NO DMC'S OR ILLEGAL DMC'S SELECTED
290 000320 050677 000700 RESTR1: CLR FIRST ;CLEAR FIRST TIME FLAG
291 000330 012700 003324 LOOP: CLR #RBUF11,R0 ;GET SET TO CLEAR BUFFERS
292 000330 005020 CLR #R0,+ ;CLEAR BUFFER
293 000336 022700 005124 1$: CMP #BASE1,R0 ;END OF BUFFERS?
294 000342 001374 BNE 1$ ;BR IF NO
295 000344 016700 177660 MOV SELECT,R0 ;RO=ACTIVE BITS
296 000350 001763 BEQ DROP ;DROP MODULE IF NO DEVICES ARE SLECTED
297 000352 016701 177430 MOV ADDR,R1 ;R1=DEVICE CSR
298 000356 016702 177426 MOV VECTOR,R2 ;R2=VECTOR
299 000362 012703 002776 MOV #INTLNK,R3 ;R3=POINTER TO INTERRUPT LINKAGE
300 000366 016767 177640 MOV FLAGB,XX11 ;SET END PASS FLAG FOR DEVICE #1
301 000372 016767 177630 MOV FLAGB,XX21 ;SET END PASS FLAG FOR DEVICE #2
302 000402 012767 003170 MOV #PIRING,INGOUT ;SET UP ALL QUEUES
303 000410 012767 003170 MOV #PIRING,INGOUT ;SET UP ALL QUEUES
304 000416 012767 003210 MOV #PIROUTQ,OUTQOUT
305 000424 012767 003210 MOV #PIROUTQ,OUTQOUT
306 000430 012767 002624 MOV #REGQ,REGQ1
307 000440 012767 002614 MOV #REGQ,REGQ1
308 000446 006200 2$: ASR R0 ;ACTIVE?
309 000450 103410 BCS 4$ ;BR IF ACTIVE
310 000454 062701 BEQ 3$ ;BR IF DONE
311 000460 062702 000010 3$: ADD #10,R2 ;UPDATE VECTOR
312 000466 062703 000034 ADD #34,R3 ;UPDATE LINK
313 000470 062703 BR 2$ ;CONTINUE
314 000472 010312 4$: MOV R3,(R2) ;LOAD VECTOR
315 000474 116762 177312 MOVB BR1,2(R2) ;LOAD INTERRUPT LEVEL
316 000476 101633 000010 MOV R1,10(R3) ;LOAD CSR TO LINKAGE
317 000506 013633 000004 MOV #R4,4(R2) ;LOAD LINKAGE ADDRESS IN VECTOR
318 000512 062762 000004 ADD #4,4(R2) ;ADJUST IT
319 000520 116762 177266 MOVB BR1,0(R2) ;LOAD INTERRUPT LEVEL
320 000526 005020 CLR 20(R3) ;CLEAR INPUT OFFSET LOCATION
321 000536 005063 000020 CLR 12(R3) ;CLEAR OUTPUT COUNT LOCATION
322 000542 005063 000012 CLR 12(R3) ;CLEAR END PASS FLAG
323 000542 005767 177466 TST FIRST ;BEGINNING OF A PASS?
324 000550 005063 BNE 5$ ;BR IF NOT
325 000554 005063 000024 CLR 26(R3) ;CLEAR AREA TO SAVE
326 000560 005063 000030 CLR 30(R3) ;CLEAR AREA ERROR
327 000566 000530 000032 CLR 32(R3) ;COUNTS FOR COMPARISON
328 000570 000530 BR ;CONTINUE
329 000572 005263 000012 5$: INC 12(R3) ;SET BIT0 OF ENDPASS FLAG
330 000576 000726 BR 3$ ;CONTINUE
331 000600 016701 177202 SETUP2: MOV ADDR,R1 ;R1=DEVICE CSR
```

```
334 000604 016700 177420 MOV SELECT,R0 ;RO=ACTIVE BITS
335 000610 006200 1$: ASR R0 ;ACTIVE?
336 000612 103404 BCS 3$ ;BR IF YES
337 000614 001454 BEQ SCAN ;BR IF DONE
338 000616 062701 000010 2$: ADD #10,R1 ;UPDATE CSR
339 000622 000772 BR 1$ ;CONTINUE
340 000624 005767 177404 3$: TST FIRST ;FIRST PASS?
341 000630 001002 BNE 4$ ;MASTER CLEAR FIRST TIME ONLY
342 000632 012711 040000 MOV #BIT14,(R1) ;MASTER CLEAR
343 000636 005711 4$: TST (R1) ;RUN SET?
344 000640 100415 BMI 5$ ;BR IF YES
345 000642 010067 MOV #SAR0,R0 ;SAVE RO
346 000646 010167 MOV #SAR1,R1 ;SAVE R1
347 000652 104407 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
348 000656 104407 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
349 000662 016700 MOV SAR0,R0 ;RESTORE RO
350 000666 016701 MOV SAR1,R1 ;RESTORE R1
351 000672 000761 BR 4$ ;WAIT FOR RUN
352 000674 052761 000100 000002 5$: BIS #100,2(R1) ;SET IE0
353 000676 032761 000001 177106 BIT #BIT0,SRI ;IS SRI ZERO?
354 000710 001003 BNE 7$ ;BR IF NO (TURNAROUND CONNECTOR)
355 000712 052711 004000 BIS #4000,(R1) ;OTHERWISE SET LU LOOP
356 000716 000402 BR 7$ ;CONTINUE
357 000720 042714 6$: BIC #4000,(R1) ;IF SRI BIT0 IS SET,CLEAR LU LOOP
358 000724 005767 177304 7$: TST FIRST ;FIRST TIME HERE?
359 000730 001003 BNE 8$ ;BR IF NO
360 000732 052711 BIS #143,(R1) ;SET LU LOOP, IEI,RQI,BASEI
361 000736 000736 BR 7$ ;CONTINUE NEXT DEVICE
362 000740 052711 000144 8$: BIS #144,(R1) ;SET LU LOOP IEI,REC BA/CC
363 000744 000724 BR 2$ ;CONTINUE NEXT DEVICE
```



```

364 ;SCAN ALL ENDPASS FLAGS UNTIL
365 ;ALL ACTIVE DMC11S ARE FINISHED
366 ; (ENDPASS FLAG = 37)
367
368
369 000746 012767 000003 177262 SCAN: MOV #3,MASK ;SET BIT FOR ALL DEVICES
370 000754 012767 000010 177242 MOV #0,DLY1 ;DELAY COUNT
371 000762 012767 177240 CLR DLY2 ;CLEAR DLY2
372 000766 012767 002014 1$: CMP FLAG,XX11 ;DEVICE 1 DONE?
373 000774 011003 BNE 25 ;BR IF NO
374 000776 042767 000001 177232 BIC #1,MASK ;DEVICE 1 IS DONE SO CLEAR BIT0
375 001004 012767 177222 002032 CMP FLAG,XX21 ;DEVICE 2 DONE?
376 001012 011003 BNE 35 ;BR IF NO
377 001014 042767 000002 177214 BIC #2,MASK ;DEVICE 2 IS DONE SO CLEAR BIT1
378 001022 005767 177210 TST MASK ;ARE ALL DEVICES FINISHED?
379 001026 011146 BNE 165 ;BR IF NO
380 001030 012701 003006 MOV #INTLMK+10,R1 ;R1 IS POINTER TO DEVICE CSR
381 001034 016700 MOV SELECT,R0 ;R0 CONTAINS BITS FOR ACTIVE DEVICES
382 001040 012703 003264 4$: CMP #BUFTAB,R3 ;R3 IS POINTER TO DEVICE RECEIVER BUFFER
383 001044 006200 ASR R0 ;ACTIVE?
384 001046 103414 BCS 65 ;BR IF YES
385 001050 001404 BEQ 65 ;BR IF DONE
386 001052 062701 000034 5$: ADD #34,R1 ;UPDATE R1 TO NEXT DEVICE CSR
387 001056 013367 TST (R3)+ ;UPDATE R3 TO NEXT BUFFER
388 001060 000771 BR 45 ;CONTINUE
389 001062 012767 177777 177144 6$: MOV #1,FIRST ;SET FIRST FLAG TO -1
390 001070 104413 ENDDIT$,BEGIN ;MONITOR SHALL TEST END OF PASS
391
392 001074 000167 177230 7$: JMP LOOP ;LOOP MODULE
393 001100 012767 000007 177130 8$: MOV #7,MASK ;MASK = BUFFER COUNT
394 001106 012704 R2,R2 ;R2 POINTS TO FIRST REC BUFFER
395 001110 012705 003066 9$: MOV #XBUF,R4 ;R4 POINTS TO RECEIVER DATA
396 001112 012705 003066 MOV #XBUF,R5 ;R5 POINTS TO GOOD DATA
397 001116 121514 10$: CMPB (R5),(R4) ;COMPARE DATA
398 001120 001414 BEQ 115 ;BR IF GOOD
399 001122 010567 MOV R5,SBADR ;LOAD GOOD ADDRESS
400 001126 010567 MOV R4,WASADR ;LOAD BAD ADDRESS
401 001132 010467 MOVB (R5),ASB ;LOAD GOOD DATA
402 001136 111467 MOVB (R4),WAS ;LOAD BAD DATA
403 001142 111467 ;*****DATA*****
404
405 001146 104404 000000 DATERS,BEGIN ;DATA ERROR!!!
406 ;*****
407 001152 122524 11$: CMPB (R5),(R4)+ ;POP DATA POINTERS
408 001154 122115 BNE #TERM,(R5) ;DONE YET?
409 001160 001356 DEC 105 ;BR IF NO
410 001162 005350 DEC #3 ;DEC BUFFER COUNT
411 001166 005350 BNE 95 ;BR IF NOT ALL 7 BUFFERS CHECKED
412 001170 012705 000003 MOV #3,R5 ;NOW LETS CHECK BASE TABLE ERROR COUNTS
413 001174 016404 000006 MOV 6(R1),R4 ;GET BASE TABLE ADDRESS
414 001178 060502 ADD R5,R2 ;GET OFFSET TO ERROR COUNTS
415 001182 010567 MOV R5,R2 ;GET POINTER TO CSR
416 001204 062702 000014 ADD #14,R2 ;MAKE IT POINT TO SAVED ERROR COUNTS
417 001210 122224 12$: CMPB (R2),(R4)+ ;COMPARE BASE TABLE ERROR COUNTS
418 001214 005202 BNE R5 ;TO SAVED ERROR COUNTS BR IF NOT SAME
419 INC R5 ;BUMP TO NEXT ERROR COUNT

```

```

420 001216 022705 000013 CMP #13,R5 ;ALL DONE YET?
421 001222 001372 BNE 125 ;BR IF NO
422 001226 000712 BR 55 ;CONTINUE NEXT DEVICE
423
424 001226 104403 000000 001576 13$: MSGNS,BEGIN,SOFT ;ASCII MESSAGE CALL WITH COMMON HEADER
425 001234 012705 000003 MOV #3,R5 ;BASE OFFSET TO ERROR COUNTS
426 001240 012701 001504 MOV #R1,R2 ;LOAD COUNTS FOR TYPEOUT
427 001244 016104 000006 MOV 6(R1),R4 ;GET BASE ADDRESS
428 001250 060504 ADD R5,R4 ;ADD IN OFFSET
429 001252 112422 14$: MOVB (R4),(R2)+ ;OK LOAD TABLE FOR TYPEOUT
430 001256 010567 INC R2 ;INCREMENT COUNTER
431 001262 022705 000013 CMP #13,R5 ;DONE YET?
432 001262 001372 BNE 145 ;BR IF NOT
433 ;SAVE THESE BASE TABLE ERROR COUNTS
434 MOV #14,R2 ;GET POINTER TO CSR
435 001266 062702 000014 ADD #14,R2 ;MAKE IT POINT TO SAVED ERROR COUNTS
436 001272 012704 001602 MOV #ESAV1,R4 ;R4 POINTS TO NEW COUNT VALUES
437 001276 012422 15$: CMP #ESAV4+2,R4 ;STORE BASE TABLE COUNTS
438 001300 010567 BNE 155 ;BR IF NOT
439 001304 001374 MOV (R1),CSRA ;LOAD CSR
440 001306 011167 MOV (R1),R5 ;LOAD CSR
441 001312 011105 MOV (R5),ASR ;SAV CONTENTS OF SEL0
442 001320 016167 176562 MOV 6(R1),ASR ;SAVE BASE ADDRESS
443 001326 012767 000001 176552 MOV #1,ERRTYP ;DATA ERROR
444 ;*****
445 ;*****
446 001334 104406 000000 001626 16$: S0PERS,BEGIN,FTABLE ;BASE TABLE DDCMP ERROR COUNTERS
447 ;*****
448 001342 000643 BR 55 ;CONTINUE NEXT DEVICE
449
450 001344 104407 000000 16$: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
451 001350 104407 000000 ;THEN CONTINUE AT NEXT INSTRUCTION.
452 001354 005367 176646 DEC DLY2 ;DEC DELAY COUNT
453 001360 001402 BEQ 16 ;BR IF NOT DONE
454 001362 000167 177400 JMP 16 ;DEC DELAY COUNT
455 001366 005367 176632 DEC DLY1 ;BR IF NOT DONE
456 001372 001402 BEQ 16 ;DEC DELAY COUNT
457 001374 000167 177366 JMP 16 ;BR IF NOT DONE
458 001400 016700 176632 MOV MASK,R0 ;PUT BITS OF HUNG DEVICES IN R0
459 001404 040067 176620 BIC R0,SELECT ;DROP ANY HUNG DEVICES
460 001410 006000 ROR R0 ;WAS DEVICE 1 HUNG?
461 001412 103004 BCC 175 ;BR IF NOT
462 001414 004367 JSR R3,XERR ;TYPE ERROR MESSAGE
463 001420 003006 CSR1 ;POINTER TO DEVICE #1 CSR
464 001422 000001 1 ;DEVICE NUMBER FOR TYPEOUT
465 001424 006000 ROR R0 ;WAS DEVICE 2 HUNG?
466 001426 103004 BCC 175 ;BR IF NOT
467 001430 004367 JSR R3,XERR ;TYPE ERROR MESSAGE
468 001434 003042 CSR2 ;POINTER TO DEVICE #2 CSR
469 001436 000002 2 ;DEVICE NUMBER FOR TYPEOUT
470 001440 000167 176664 18$: JMP LOOP ;RESTART MODULE
471
472 001444 012302 XERR: MOV (R3)+,R2 ;GET POINTER TO CSR
473 001446 012367 000112 MOV (R3)+,DEV ;GET DEVICE NUMBER
474 001448 002767 000060 BIC #6,DEV ;MAKE IT ASCII
475 001460 104403 000000 001566 MSGNS,BEGIN,DROP1 ;ASCII MESSAGE CALL WITH COMMON HEADER

```

```

476 001466 011201          MOV      (R2),R1          ;GET CSR ADDRESS
477 001470 011187          MOV      R1,CSRA         ;SAVE CSR
478 001474 011187          MOV      (R1),ACSR      ;SAVE CONTENTS OF SEL0
479 001500 016167          MOV      2(R1),ASTAT    ;SAVE CONTENTS OF SEL2
480 001506 016167          MOV      4(R1),DLV1     ;SAVE CONTENTS OF SEL4
481 001512 016167          MOV      6(R1),DLV2     ;SAVE CONTENTS OF SEL6
482 001522 016167          MOV      8(R2),ESAV1    ;END PASS FLAG
483 001530 016267          MOV      10(R2),ESAV2   ;RECEIVE BUFFER OFFSET
484 001536 016267          MOV      12(R2),ESAV3   ;RECEIVE BUFFER COUNTERS
485 001544 012767          MOV      14(R2),ESAV4   ;NO INTERRUPT
486                                     ;*****
487 001552 104405 000000 001612  HRDERS BEGIN ETABLE     ;DUMP DMC CSR'S AND STATUS FLAGS
488                                     ;*****
489 001560 005011          CLR      (R1)           ;SHUT OFF HUNG DMC11
490 001562 000203          RTS      R3             ;RETURN
491
492 001564 000000          DEV:      0
493
494 001566 001640          DROP1:   XDROP1
495 001570 001564          DEV
496 001572 001661          XDROP2
497 001574 177777          -1
498
499 001576 001717          SOFT1:   SOFT1
500 001600 177777          -1
501
502 001602 000000          ESAV1:   0              ;EXTENDED ERROR PRINTOUT LOCATIONS
503 001604 000000          ESAV2:   0
504 001606 000000          ESAV3:   0
505 001610 000000          ESAV4:   0
506
507 001612 000224          ETABLE:  DLV1           ;TABLE OF ADDRESSES FOR EXTENDED ERROR PRINTOUT
508 001614 000226          DLV2
509 001616 001602          DLV3
510 001620 001604          ESAV1
511 001622 001606          ESAV2
512 001624 177777          -1
513
514 001626 001602          FTABLE:  ESAV1
515 001630 001604          ESAV2
516 001632 001606          ESAV3
517 001634 001610          ESAV4
518 001636 177777          -1
519
520 001640 042045 041515 030461  XDROP1:  .ASCIZ  %DMC11 DEVICE # /
521 001646 042045 053105 041511
522 001654 020105 020043 000
523 001661 040 051511 044040  XDROP2:  .ASCIZ  / IS HUNG AND HAS BEEN DROPPED/
524 001666 047123 020107 047101
525 001674 020103 040510 020123
526 001702 042502 047105 042040
527 001710 047522 050120 042105
528 001716 040
529 001718 000 047523 052106  SOFT1:   .ASCIZ  %SOFT ERROR - DDCMP ERROR COUNTERS ARE NON ZERO%/
530 001724 042440 051122 051117
531 001732 026440 042040 041504

```

```

532 001740 050115 042440 051122
533 001746 051117 041440 052517
534 001754 052116 051105 020123
535 001762 051101 020105 047516
536 001770 020116 042532 047522
537 001776 000045
538
.EVEN

```

```

539 002000 010577 001244 IISR: MOV R5, @INQIN ;STORE LINK POINTER IN QUEUE
540 002004 062767 000000 ADD #2, INQIN ;UPDATE QUEUE
541 002017 022767 003210 CMP #PIRINGQ+20, INQIN ;END OF QUEUE?
542 002026 001003 BNE #3 ;BR IF NO
543 002022 012767 003170 001220 1$: MOV #PIRING, INQIN ;RESET QUEUE POINTER
544 002030 012605 MOV (SP)+, R5 ;RESTORE R5
-----
545 002032 000004 000000 002040 PIRQS, BEGIN, 2$ ; QUEUE UP TO CONTINUE AT 2$ AND RTI
546
547
548 002040 017705 001206 2$: MOV @INQOUT, R5 ;GET LINK POINTER IN R5
549 002044 022767 000000 ADD #2, INQOUT ;UPDATE QUEUE
550 002052 003210 001170 CMP #PIRINGQ+20, INQOUT ;END OF QUEUE?
551 002060 001003 BNE #3 ;BR IF NO
552 002066 012767 003170 001162 3$: MOV #PIRING, INQOUT ;RESET QUEUE POINTER
553 002070 015561 000004 MOV #4, (R1) ;LOAD CSR ADDRESS
554 002074 032711 000007 BIT #4, (R1) ;XMIT BA/CC?
555 002100 001511 BEQ #XMIT ;BR IF YES
556 002106 032711 000004 BIT #4, (R1) ;REC BA/CC?
557 002106 032711 000004 BNE #4, (R1) ;BR IF YES
558 002110 032711 000002 BIT #2, (R1) ;CNTL I?
559 002114 001424 BEQ #4 ;BR IF YES
560 002116 016567 000012 176120 MOV #2, (R5), VA ;LOAD VA WITH VIRTUAL ADDRESS
561 002122 001424 JSR #4, (R1) ;GET PHYSICAL ADDRESS
562 002130 016761 176112 000004 MOV #6, (R1) ;LOAD PHYSICAL BASE ADDRESS
563 002136 016761 176106 000006 MOV #6, (R1) ;LOAD EA BITS FOR BASE ADDRESS
564 002144 004767 000260 JSR #PC, ENDCLR ;CLEAR ROI
565 002150 052765 000001 000006 BIS #1, 6(R5) ;SHOW THAT A BASE ADDRESS WAS LOADED
566
567 002156 152711 000004 4$: BISS #41, (R1) ;ASK FOR CNTL I
568 002162 104400 000000 EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
569 002166 004767 000004 CLR #6, (R1) ;SET FULL DUPLEX
570 002172 004767 000023 JSR #PC, ENDCLR ;CLEAR ROI
571 002176 152711 000044 BISS #44, (R1) ;ASK FOR REC BA/CC I
572 002202 104400 000000 EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
573 002206 016567 000010 REC: MOV #10, (R5), R4 ;GET REC BUFFER POINTER
574 002212 066504 000014 ADD #14, (R5), R4 ;ADD OFFSET
575 002216 011467 176022 MOV #4, (R4), VA ;VA GETS REC BUFFER VIRTUAL ADDRESS
576 002222 004767 005614 JSR #PC, #ABITS ;GET PHYSICAL ADDRESS
577 002226 016761 176014 000004 MOV #4, (R1) ;LOAD PHYSICAL RBUF
578 002234 016761 176010 000006 MOV #6, (R1) ;LOAD RBUF EA BITS
579 002242 056761 175772 000006 BIS #COUNT, 6(R1) ;LOAD RECEIVER COUNT
580 002250 004767 000153 JSR #PC, ENDCLR ;CLEAR ROI
581 002254 004767 000153 ADD #4, (R5) ;UPDATE OFFSET
582 002262 022765 000016 000014 CMP #16, 14(R5) ;LOADED 7 BUFFERS YET?
583 002270 001404 BEQ #1 ;BR IF YES
584 002274 001404 BISS #4, (R1) ;REQUEST REC BA/CC
585 002278 104400 000000 EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
586 002302 052765 000002 000006 1$: BISS #2, 6(R5) ;SET BIT2 IN END PASS FLAG TO
587 ;SHOW THAT ALL 7 REC BA/CC WERE LOADED
588
589 002310 005065 000014 CLR #14, (R5) ;CLEAR OFFSET
590 002314 152711 000040 BISS #40, (R1) ;ASK FOR XMIT BA/CC I
591 002320 104400 000000 EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
592 002324 012767 003066 175712 XMIT: MOV #XBUF, VA ;LOAD XBUF VIRTUAL ADDRESS
593 002328 004767 000104 JSR #PC, #ABITS ;GET PHYSICAL ADDRESS
594 002336 012761 175704 000004 MOV #4, (R1) ;LOAD PHYSICAL XBUF
595 002344 016761 175700 000006 MOV #6, (R1) ;LOAD EA BITS

```

```

595 002352 056761 175664 000006 BIS #TCOUNT, 6(R1) ;LOAD XMIT COUNT
596 002360 004767 000044 JSR #PC, ENDCLR ;CLEAR ROI
597 002364 005265 000014 INR #14, (R5) ;COUNT HOW MANY XMIT BA/CC HAVE BEEN LOADED
598 002370 022765 000007 000014 CMP #7, 14(R5) ;DO WE HAVE ALL 7?
599 002376 001404 BEQ #1 ;BR IF YES
600 002400 152711 000040 BISS #40, (R1) ;ASK FOR XMIT BA/CC
601 002404 104400 000000 EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
602 002410 052765 000004 000006 1$: BISS #4, 6(R5) ;SET BIT2 IN END PASS FLAG TO
603 ;SHOW THAT ALL 7 XMIT BA/CC WERE LOADED
604 002416 005065 000014 CLR #14, (R5) ;CLEAR OFFSET
605 002422 105011 000000 CLR #1, (R1) ;CLEAR I
606 002424 104400 000000 EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
607
608 002430 142711 000040 ENDCLR: BICB #40, (R1) ;CLEAR ROI
609 002434 105711 000040 1$: TSTB #1, (R1) ;IS ROI GONE?
610 002436 100776 BMI #1 ;BR IF NO
611 002440 000207 RTS #PC ;RETURN
612
613 002442 000000 000000 000244 EABITS: GETPAS, BEGIN, VA ;GET PHYSICAL ADDRESS FROM 16-BIT VA
614 002442 104415 000000 SWAB #EA ;BITS 4+5 TO 13+12
615 002450 000367 175574 ROL #EA ;NOW 14+13
616 002454 006167 175570 ROL #EA ;NOW 15+14
617 002460 006167 175564 BIC #37776, EA ;CLEAR ALL BUT 14 & 15
618 002464 042767 037776 175556 BIC #37776, EA ;CLEAR ALL BUT 14 & 15
619 002472 000207 RTS #PC ;RETURN
620

```

```

621 002474 010577 000554 DISR: MOV R5,OUTQIN ;MOVE LINK POINTER TO QUEUE
622 002500 062767 000002 ADD #2,OUTQIN ;UPDATE QUEUE
623 002500 062767 003230 000540 CMP #PIROUTQ+20,OUTQIN ;END OF QUEUE?
624 002514 001003 BNE JS ;BR IF NO
625 002516 012767 003210 000530 MOV #PIROUTQ,OUTQIN ;RESET QUEUE POINTER
626 002524 012605 MOV (SP)+,R5 ;RESTORE R5
-----
628 002526 000004 000000 002534 PIRQS,BEGIN,2S ;QUEUE UP TO CONTINUE AT 2S AND RTI
-----
629
630 002534 017705 000516 2S: MOV OUTQOUT,R5 ;GET LINK POINTER FROM QUEUE
631 002540 062767 000002 ADD #2,OUTQOUT ;UPDATE QUEUE
632 002546 022767 003230 000502 CMP #PIROUTQ+20,OUTQOUT ;END OF QUEUE?
633 002554 001003 BNE JS ;BR IF NO
634 002556 012767 003210 000472 MOV #PIROUTQ,OUTQOUT ;RESET QUEUE POINTER
635 002564 011501 MOV (R5),R1 ;LOAD CSR ADDRESS
636 002566 032761 000001 000002 BIT #BIT0,2(R1) ;ERROR?
637 002574 001422 BEQ JS ;BR IF NO
638 002576 018167 MOV R1,CSRA ;LOAD DEVICE CSR
639 002600 016167 MOV 4(R1),ACSR ;LOAD CONTENTS OF DEVICE CSR
640 002610 016167 000004 175272 MOV 6(R1),ASTAT ;LOAD ERROR BITS
641 002616 005067 175264 CLR ERRTRY6 ;UNKNOWN ERROR
-----
643 002622 104405 000000 000000 *****
644 HRDRS,BEGIN,NULL ;A CNTL 0 WAS RECEIVED,ASTAT=ERROR BITS
645 *****
646 002630 142761 000207 000002 BIC #207,2(R1) ;CLEAR RDO
647 002636 032761 000000 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
648 002642 032761 000004 000002 4S: BIT #BIT2,2(R1) ;RECEIVER DONE?
649 002650 001025 BNE JS ;BR IF YES
650 002652 142761 000207 000002 BIC #207,2(R1) ;CLEAR RDO
651 002658 032761 000013 000012 INCB 13(R5) ;LO BYTE IS XMIT DONE COUNT
652 002664 010101 000007 000012 CMPB #7,13(R5) ;DO WE HAVE 7 XMIT DONES YET?
653 002674 052765 000010 000002 SNE JS ;BR IF NO
654 BIS #10,2(R5) ;SET BIT3 IN ENDPASS FLAG TO
655 ;SHOW THAT WE GOT 7 XMIT DONES
656 002702 026765 175324 000002 CMP FLAGB,2(R5) ;ALL DONE?
657 002710 001003 BNE JS ;BR IF NO
658 002712 042761 000100 000002 BIC #100,2(R1) ;CLEAR OIE IF YES
-----
659 002720 104400 000000 000000 5S: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
660 002724 142761 000207 000002 6S: BIC #207,2(R1) ;CLEAR RDO
661 002732 052765 000013 000012 INCB 13(R5) ;HI BYTE IS REC DONE COUNT
662 002744 010101 000007 000013 CMPB #7,13(R5) ;DO WE HAVE 7 REC DONES YET?
663 002746 052765 000020 000002 SNE JS ;BR IF NO
664 BIS #20,2(R5) ;SET BIT4 IN ENDPASS FLAG TO
665 ;SHOW THAT WE GOT ALL 7 REC DONES
666 002754 026765 175252 000002 CMP FLAGB,2(R5) ;ALL DONE?
667 002762 001003 BNE JS ;BR IF NO
668 002764 042761 000100 000002 BIC #100,2(R1) ;CLEAR OIE IF YES
-----
669 002772 104400 000000 7S: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
670 002772 104400 000000

```

```

671 ;LINK TABLE TO INTERRUPT SERVICE ROUTINES
672 ;-----
673
674 002776 004567 176776 INTLNK: JSR R5,IISR
675 003002 004567 177466 JSR R5,DISR
676
677 003006 000000 CSR1: .WORD ;DMC CSR FOR DEVICE 1
678 003010 000000 XX11: .WORD ;END PASS FLAG FOR DEVICE 1
679 003014 052770 .WORD ;RECEIVE BUFFER POINTER FOR DEVICE 1
680 003018 000000 .WORD ;BASE ADDRESS FOR DEVICE 1
681 003016 000000 XX12: .WORD ;RECEIVE BUFFER OFFSET FOR DEVICE 1
682 003020 000000 .WORD ;REC/XMIT COUNTERS
683 003022 000000 .WORD ;THESE NEXT 8 BYTES ARE FOR
684 003024 000000 .WORD ;THE DDCMP ERROR COUNTS
685 003026 000000 .WORD ;IN THE BASE TABLE TO BE
686 003030 000000 .WORD ;SAVED FOR COMPARISION.
687
688 003032 004567 176742 JSR R5,IISR
689 003036 004567 177432 JSR R5,DISR
690
691 003042 000000 CSR2: .WORD ;DMC CSR FOR DEVICE 2
692 003044 000000 XX21: .WORD ;END PASS FLAG FOR DEVICE 2
693 003046 003306 .WORD ;RECEIVE BUFFER POINTER FOR DEVICE 2
694 003050 005274 .WORD ;BASE ADDRESS FOR DEVICE 2
695 003052 000000 XX22: .WORD ;RECEIVE BUFFER OFFSET FOR DEVICE 2
696 003054 000000 .WORD ;REC/XMIT COUNTERS
697 003056 000000 .WORD ;THESE NEXT 8 BYTES ARE FOR
698 003060 000000 .WORD ;THE DDCMP ERROR COUNTS
699 003062 000000 .WORD ;IN THE BASE TABLE TO BE
700 003064 000000 .WORD ;SAVED FOR COMPARISION.

```

```
701 ;BUFFERS & QUEUES
702 ;-----
703
704 003066* 001001 004004 020020 XBUF: .ASCII <001<002<004<010<020<040<100<200<377<376>
705 003074* 001000 177777 137737 .ASCII <375<373<367<357<337<277<177<037<076<174>
706 003100* 175775 167767 137737 .ASCII <370<360<001<340<003<300<007>/ABCDEFGHJKLM/
707 003106* 017577 076076
708 003112* 170370 160001 140003
709 003126* 052524 041502 042504 .ASCIIZ /NOPQRSTUWXYZ01234567890/
710 003126* 043506 044510 045512
711 003134* 046514
712 003136* 047516 050520 051522
713 003144* 052524 053526 054530
714 003152* 030132 031061 032063
715 003160* 033065 034067 030071
716 003166* 000
717 003170*
718 .EVEN
719 003170* 000010 PIRING: -BLKW 10
720 003210* 000010 PIROUTQ: -BLKW 10
721 003230* 000010 REGQ: -BLKW 10
722 003250* 000000 INGIN: 0
723 003252* 000000 INQOUT: 0
724 003254* 000000 OUTQIN: 0
725 003256* 000000 OUTQOUT: 0
726 003260* 000000 REGQ1: 0
727 003262* 000000 REGQ0: 0
728
729 ;TABLE OF RECEIVE BUFFER POINTERS
730
731 003264* 003270* RBUF1: RBUF1 ;BUFFER POINTER FOR DEVICE 1
732 003266* 003306* RBUF2: RBUF2 ;BUFFER POINTER FOR DEVICE 2
733
734 ;TABLE OF RECEIVE BUFFERS
735
736 003270* RBUF1: RBUF1 ;RECEIVE BUFFERS FOR DEVICE 1
737 003270* 003324* RBUF11
738 003272* 003424* RBUF12
739 003274* 003524* RBUF13
740 003276* 003624* RBUF14
741 003300* 003724* RBUF15
742 003302* 004024* RBUF16
743 003304* 004124* RBUF17
744
745 003306* 004224* RBUF2: RBUF2 ;RECEIVE BUFFERS FOR DEVICE 2
746 003310* 004324* RBUF21
747 003312* 004424* RBUF22
748 003314* 004524* RBUF23
749 003316* 004624* RBUF24
750 003320* 004724* RBUF25
751 003322* 005024* RBUF26
752
753 ;RECEIVE BUFFERS FOR DEVICE 1
754
755
756
```

```
757 003324* 000100 RBUF11: -BLKB 100 ;RECEIVE BUFFER 11
758 003424* 000100 RBUF12: -BLKB 100 ;RECEIVE BUFFER 12
759 003524* 000100 RBUF13: -BLKB 100 ;RECEIVE BUFFER 13
760 003624* 000100 RBUF14: -BLKB 100 ;RECEIVE BUFFER 14
761 003724* 000100 RBUF15: -BLKB 100 ;RECEIVE BUFFER 15
762 004024* 000100 RBUF16: -BLKB 100 ;RECEIVE BUFFER 16
763 004124* 000100 RBUF17: -BLKB 100 ;RECEIVE BUFFER 17
764
765 ;RECEIVE BUFFERS FOR DEVICE 2
766
767 004224* 000100 RBUF21: -BLKB 100 ;RECEIVE BUFFER 21
768 004324* 000100 RBUF22: -BLKB 100 ;RECEIVE BUFFER 22
769 004424* 000100 RBUF23: -BLKB 100 ;RECEIVE BUFFER 23
770 004524* 000100 RBUF24: -BLKB 100 ;RECEIVE BUFFER 24
771 004624* 000100 RBUF25: -BLKB 100 ;RECEIVE BUFFER 25
772 004724* 000100 RBUF26: -BLKB 100 ;RECEIVE BUFFER 26
773 005024* 000100 RBUF27: -BLKB 100 ;RECEIVE BUFFER 27
774
775
776 005124* 000400 BASE1: -BLKB 256. ;BASE TABLE FOR DEVICE 1
777
778 005524* 000400 BASE2: -BLKB 256. ;BASE TABLE FOR DEVICE 2
779
780
781
782 000001 .END
```


RBUF12	003424R	738	758#											
RBUF13	003524R	739	759#											
RBUF14	003624R	740	760#											
RBUF15	004024R	742	763#											
RBUF16	004124R	743	763#											
RBUF17	004154R	743	763#											
RBUF21	003306R	692	732	744#										
RBUF22	004224R	742	763#											
RBUF23	004424R	747	769#											
RBUF24	004524R	748	770#											
RBUF25	004724R	750	773#											
RBUF26	004754R	750	773#											
RBUF27	005024R	751	773#											
RRCOUNT	000240R	267#	579											
RFC	002206R	557	577#											
RFGO	004230R	306	307#	721#										
RFGQI	003260R	306*	307#											
RFGQO	003262R	307*	307#											
RFSVRT	000324R	253	287	290#										
RESS1	000056R	237#	237#											
RESS2	000060R	237#	237#											
RSTRT	000112R	493												
SARO	000248R	470	345*	349										
SARL	000254R	470	346*	350										
SADR	000102R	246#	400*											
SCAN	000746R	337#	369#											
SELECT	000230R	418	433#	295	334	381	459*							
SETOP2	000600R	418	433#											
SOPCNT	000042R	229												
SOPERS	104406	460	446											
SOPFAS	000048R	231												
SOPF	001576R	424	499#											
SOPT1	001717R	469	529#											
SPOINT	000032R	227												
SPSIZ	000040R	231	258											
SR1	000016R	218	353											
SR2	000020R	219												
SR3	000022R	220												
SR4	000024R	221												
START	000256R	424	281#											
STAT	000026R	223												
SVRO	000062R	438												
SVR1	000066R	440												
SVR2	000066R	440												
SVR3	000070R	441												
SVR4	000072R	442												
SVR5	000074R	443												
SVR6	000076R	444												
SVSCNT	000052R	233												
TCOUNT	000028R	213	595											
TERM	000608	470	408											
TRPDFD	000022	260												
VA	000244R	271	575*	591*	614									
VECTOR	000010R	248	298											
WASADR	000104R	248	401*											

WDFR	000116R	255#	283*											
WDTD	000114R	254#	282*											
XBUF	003066R	396	591	704#										
XDRDP1	001640R	496	520#											
XDRDP2	001661R	496	523#											
XERR	001444R	462	467	472#										
XFLAG	000005R	212												
XMIT	002322R	552	591#											
XX11	003010R	300*	372	678#										
XX13	003016R	681												
XX21	003044R	681	375	691#										
XX22	003052R	694												
XX23	003054R	695												
.	006124R	767#	456	717#	719#	720#	721#	757#	758#	759#	760#	761#	762#	763#
			768#	769#	770#	771#	772#	773#	774#	775#				

. ABS. 000000 000
 006124 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 XDMCBO XDMCBO/SQL/CRF:SYM=DDXCOM,XDMCBO
 RUN-TIME: 23.4 SECONDS
 RUN-TIME RATIO: 14/6=2.2
 CORE USED: 11K (21 PAGES)

DIAGNOSTIC ENGINEERING



DECO DEPO SUBMISSION

NEW

FOR RELEASE ENG. USE

CHANGE DELETE

PRODUCT IDENTIFICATION

MD	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO FULLY	PRODUCT DATE			STATUS	DISTRIBUTION		1ST COPY - RIGHT YEAR	LAST COPY - RIGHT YEAR	
	ZZ	CXDMC	B	1	01	24	APR	79	<input type="checkbox"/> OBSOLETE	X	G		R	1976

TITLE CXDMCB1 DMC-11 MODULE
 AUTHOR D. BUTENHOF MAINTENANCE/PT SUPT GRP MAINTAINER D. BUTENHOF SUBMITTING ENGINEER D. BUTENHOF

PRODUCT COMPONENTS

CK	DESCRIPTION	PRODUCT NO.	REV	CK	DESCRIPTION	PRODUCT NO.	REV
	DOCUMENT				INDEX		
	LISTING				SOURCE MEDIA		
	OBJECT MEDIA				TEST MEDIA		
		AF-E953B-M1					

PRODUCTS OBSOLETE (other than previous version)

LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV
MD			MD			MD		

PRODUCT CHARACTERISTICS

PROCESSORS PRODUCT OPERATES WITH (Enter all applicable 2-digit codes representing the Processor the product operates with. See separate instructions.)
 OPERATIONAL CODES (Enter all applicable 2-digit codes that describe the product. See separate instructions.)
 ACT/APT/XXDP EXT ACT SEQ NUMBER ACT/XXDP COMPATIBLE? APT COMPATIBLE? 1ST PASS RUN TIME SUBSEQUENT PASS RUN TIME
 INFORMATION FIELD Y N Y N SECONDS SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED: _____
 DEVICE AFFECTED _____ MULTIMEDIA AFFECTED? YES NO
 KIT NUMBERS ZJ130-RB _____
 ZJ129-RZ, FR _____

PROBLEM: THE "DDCMP ERROR COUNTERS ARE NON-ZERO" SOFT ERROR IS MISLEADING SINCE IT REFERS TO A RECOVERABLE AND SOFTWARE-TRANSPARENT SITUATION. NEVERTHELESS, IF 40 SUCH OCCUR, THE MODULE WILL BE DROPPED BY THE MONITOR.

SOLUTION: THE SOFT ERROR CALL AND MESSAGE ARE DELETED FROM THE MODULE BY THIS PATCH, AND THE HARD ERROR CALL IS EXTENDED FOR CLARIFICATION OF THE SITUATION, IF THE DMC-11 SHOULD BE UNABLE TO RECOVER.

DEPO PATCH AREA

CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO
1226	104403	402	1726	51122	51040
1334	104406	402	1730	51117	52105
1716	22400	20072	1732	26440	44522
1720	47523	53117	1734	42040	51505
1722	52106	51105	1736	41504	0
1724	42440	33440			

SUBMITTING ENGINEER <i>[Signature]</i>	MANUFACTURING ENGINEER J.E. CASPELLA	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER
DATE: 24 APR 79	DATE: 10-MAY-79	DATE:	098-05460
MAINTAINER <i>[Signature]</i>	FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. 3130
DATE:	DATE:		