

.REM >

IDENTIFICATION

PRODUCT CODE: AC-E715I-MC
PRODUCT NAME: CXDQAI0 DQ11 DEC/X11 MODULE
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) 1973,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

DQA IS AN IOMOD THAT WILL EXERCISE UP TO EIGHT DQ11S. DURING A SINGLE PASS IT WILL DO 15 CHARACTER TRANSFERS OF SEQUENTIAL DATA AND CHECK THE STATUS REGISTERS AND SECONDARY REGISTERS. IT WILL TRANSFER A 256 EIGHT BIT BINARY COUNT PATTERN 128 TIMES FOR EACH END PASS RECORDED. ANY ERRORS DETECTED DURING THE PASS ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: ONLY THE BASIC UNIT IS EXERCISED; SO A BASIC UNIT IS NEEDED. ANY EXTRA OPTIONS ON THE DQ11 WILL NOT BE EXERCISED.

STORAGE: DQA REQUIRES:
1. DECIMAL WORDS: 768
2. OCTAL WORDS: 1400
3. OCTAL BYTES: 3000

3. PASS DEFINITION

ONE PASS OF THE DQA MODULE CONSISTS OF 24576 CYCLES OF AN INCREMENTAL DATA PATTERN TRANSFERRED AT 15. CHARACTER BURSTS.

4. EXECUTION TIME.

RUNNING ALONE ON AN 11/20 ONE PASS TAKES APPROXIMATELY 20 SECONDS. NOTE: PASS TIME IS DEPENDENT UPON BAUD RATE.

5. CONFIGURATION REQUIREMENTS.

DEFAULT PARAMETERS:

DEVADR: 1 VECTOR: 1 BR1: 5 BR2: 5 DEVCNT: 1
USER MUST SPECIFY THE ADDRESS AND VECTOR OF THE FIRST DQ11 AT CONFIGURATION TIME.

6. DEVICE/OPTION SETUP

NO SPECIAL SET NECESSARY. (BASIC UNIT TESTED ONLY)

7. MODULE DESCRIPTION

- A. TESTS FOR THE AVAILABILITY OF UP TO EIGHT DQ11'S
- B. INITIALIZES ALL DQ11'S. SETS ACCORDINLY.
- C. SETS ALL GO BITS AND LEAVES MODULE
- D. GETS TRANS. INTERRUPTS. GETS ALL RECV. INTERRUPTS.
- E. CHECKS ALL STATUS REGISTERS AND SECONDARY REGISTERS
- F. CHECKS ALL DATA. REPORTS ANY ERROR FOUND.
- G. PREPARES DATA TO TRANSMITTED AGAIN.
- H. REPEATS A THROUGH G 128 X 256 CHARS.
- J. REPORTS END PASS AND CONTINUES AS ABOVE.

8. OPERATOR OPTIONS

- A. MODULE LOCATION DVID1 MAY BE CHANGED TO EXERCISE ANY COMBINATION OF DQ11S. BIT 0=DQ110
BIT 1=DQ11 1..... BIT 7=DQ11 7.
- B. IF DVID1=0 AT RUN TIME NO DQ11S WILL BE EXERCISED.

9. NON STANDARD PRINTOUTS

NONE: ALL PRINTOUTS HAVE THE STANDARD FORMAT.
IF YOU NEED HELP IN RUNNING MODULE REFER TO DEC/X11 DOCUMENT.

>

```
128 .LIST SEQ,LOC,BIN
129 IDMOD <DQAI>,1,1,5,5,7100,31
130 000000* MODULE 140000,DQAI,1,1,5,5,7100,31
131 000000* .TITLE DQAI DEC/X11 SYSTEM EXERCISER MODULE
132 ; DXDCOM VERSION 6 23-MAY-78
133 .LIST BIN
134 ;*****
135 000000* BEGIN:
136 000000* 050504 044501 040 MDDNAM: .ASCII /DQAI / ;MODULE NAME
137 000005* 000 ASCII /DQAI / ;MODULE NAME
138 000006* 000001 ADDR: 1+0 ;1ST DEVICE ADDR.
139 000010* 000001 VECTOR: 1+0 ;1ST DEVICE VECTOR.
140 000012* 240 BR1: .BYTE PRTV5+0 ;1ST BR LEVEL.
141 000013* 240 BR2: .BYTE PRTV5+0 ;2ND BR LEVEL.
142 000014* 000001 DVID1: +1 ;DEVICE INDICATOR 1.
143 000016* 000000 SR1: OPEN ;SWITCH REGISTER 1
144 000020* 000000 SR2: OPEN ;SWITCH REGISTER 2
145 000022* 000000 SR3: OPEN ;SWITCH REGISTER 3
146 000024* 000000 SR4: OPEN ;SWITCH REGISTER 4
147 ;*****
148 000026* 140000 STAT: 140000 ;STATUS WORD.
149 000030* 000224* INIT: START ;MODULE START ADDR.
150 000032* 000324* SPOINT: MODSP ;MODULE STACK POINTER.
151 000034* 000000 PASCNT: 0 ;PASS COUNTER
152 000036* 007100 ICOUNT: 7100 ;# OF ITERATIONS PER PASS=7100
153 000040* 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
154 000042* 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
155 000044* 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
156 000046* 000000 SDPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
157 000048* 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
158 000052* 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
159 000054* 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
160 000056* 000000 CNDFIG: 0 ;PRESERVED FOR MONITOR USE
161 000058* 000000 RES1: 0 ;PRESERVED FOR MONITOR USE
162 000060* 000000 RES2: 0 ;PRESERVED FOR MONITOR USE
163 000062* 000000 SVR0: OPEN ;LOC TO SAVE R0.
164 000064* 000000 SVR1: OPEN ;LOC TO SAVE R1.
165 000066* 000000 SVR2: OPEN ;LOC TO SAVE R2.
166 000070* 000000 SVR3: OPEN ;LOC TO SAVE R3.
167 000072* 000000 SVR4: OPEN ;LOC TO SAVE R4.
168 000074* 000000 SVR5: OPEN ;LOC TO SAVE R5.
169 000076* 000000 SVR6: OPEN ;LOC TO SAVE R6.
170 000100* 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
171 000102* 000000 SBADR: OPEN ;ADDR OF GOOD DATA, OR
172 000104* 000000 ACSR: OPEN ;CONTENTS OF CSR.
173 000106* 000000 WABADR: OPEN ;ADDR OF BAD DATA, OR
174 000108* 000000 ASTAT: OPEN ;STATUS REG CONTENTS.
175 000106* 000000 ERRTP: ;TYPE OF ERROR
176 000106* 000000 ASD: OPEN ;EXPECTED DATA.
177 000110* 000000 AWS: OPEN ;ACTUAL DATA.
178 000112* 000332* RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
179 000114* 000000 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
180 000116* 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
181 000120* 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
182 000122* 000031 IDNUM: 31 ;MODULE IDENTIFICATION NUMBER=31
183 .REPT SPSIZ ;MODULE STACK STARTS HERE.
```

```
184 .NLIST
185 .WORD 0
186 .LIST
187 .ENDR
188 000224* MODSP:
189 ;*****
```

```
190 000224* 012767 000010 177662 START: MOV #8,WDTO ;8 WORDS TO MEM PER ITERATION
191 000232* 012767 000010 177656 MOV #8,WDFR ;8 WORDS FROM MEM PER ITERATION
192 000240* 012767 000002 177652 MOV #2,INTR ;2 INTERRUPTS PER ITERATION
193 000246* 016700 177542 MOV #D1,R0 ;SAVE DEVICE COUNT IN R0
194 000252* 001002 BNE IS ;8R IF AT LEAST ONE SELECTED
195 000254* 104410 000000* ENDS,BEGIN ;
196 000260* 006200 1$: ASR R0 ;KILL 1ST DEV-ALREADY COUNTED
197 000262* 103476 R0 15 ;CLOCK TILL FOUND
198 000264* 006200 2$: ASR R0 ;SHIFT IN NEXT BIT
199 000266* 103011 BCC 35 ;RR IF NO DEVICE HERE
200 000270* 066767 000010 177616 ADD #8,WDTO ;DOUBLE WORDS
201 000276* 016700 000010 177612 ADD #2,WDFR ;DOUBLE WORDS FROM
202 000304* 066767 000002 177606 ADD #2,INTR ;DOUBLE INTERRUPTS
203 000312* 005700 3$: TST R0 ;ANY MORE DEVICES?
204 000314* 103476 BNE 25 ;RR IF YES
205 000316* 016767 177472 002432 MOV #D1,SELECT ;COPY THE DEVICE SELECTION PARAMETER
206 ;INTO SELECT
207 000324* 000402 3R RSTRRT ;IF ZERO, NO DEVICES SELECTED-
208 ;DROP THE MODULE
209 000326* DROP: ENDS,BEGIN ;DROP THE MODULE
210 000326* 104410 000000* RSTRRT: CLRB SELECT+1 ;ELIMINATE IRRELEVANT BITS
211 000332* 105067 002421
212
213
214 000336* 016701 002414 SETUP1: MOV SELECT,R1 ;COPY SELECT INTO R1 FOR SETUP
215 000342* 001771 B2Q ;IF SELECT WAS ZERO, GO DROP THE MODULE
216 ;IF SELECT WAS NOT ZERO, DO MODULE PROCESSING
217 000344* 016700 177440 MOV VECTOR,R0 ;LOAD R0 WITH FIRST VECTOR ADDRESS
218 000350* 016702 177432 MOV ADDR,R1 ;LOAD R1 WITH FIRST DEVICE ADDRESS
219 000354* 012703 002174* MOV #LNKTAB,R3 ;POINT R3 TO BEGINNING OF JSR LINK TABLE
220 000360* 006201 1$: ASR R1 ;ISOLATE A LINE IN THE "CM" BIT
221 000362* 103476 BCC 23 ;IF NO MORE TO BE SELECTED
222 000364* 001427 B2Q SETUP2 ;GO SET UP BUFFERS,QUEUES,ETC.
223 ;IF MORE UPDATE ALL REGISTERS..THE VECTOR
224 000366* 062700 000010 ADD #10,R2 ;THE ADDRESS OF TABLE POINTER
225 000372* 012702 ADD #16,R3 ;CHECK TO SEE IF OTHER DEVICES SELECTED
226 000376* 062703 000016 BR 15 ;LOAD THE RECEIVE ROUTINE
227 000402* 000766 2$: MOV R3,(R0)+ ;POINT R3 TO NEXT VECTOR BOUNDARY
228 000404* 010320 BR 15 ;UPDATE R0 TO NEXT VECTOR BOUNDARY
229 ;UPDATE R3 TO THE CSR INSERT LOCATION
230 000406* 116720 MOVBR BR2,(R0)+ ;LOAD THE RCV CSR INTO LINKING TABLE
231 000412* 005200 INC R0 ;UPDATE THE POINTER TO NEXT INSTRUCTION AND
232 000414* 022323 CMP R3,(R3)+ ;POINT R2 TO THE TRANSMITTER CSR
233 000416* 012703 MOV R3,(R3)+ ;LOAD THE TRANSMITTER ROUTINE
234 000420* 022322 CMP R3,(R2)+ ;POINT R2 TO THE TRANSMITTER CSR
235 ;LOAD THE TRANSMITTER ROUTINE
236 000422* 010320 MOV R3,(R0)+ ;POINT R3 TO THE TRANSMITTER CSR
237 ;LOAD THE TRANSMITTER ROUTINE
238 000424* 116720 MOVBR BR1,(R0)+ ;POINT R3 TO THE TRANSMITTER CSR
239 000430* 005200 INC R0 ;UPDATE THE VECTOR POINTER
240 000432* 022323 CMP R3,(R3)+ ;UPDATE THE LINK TABLE POINTER
241 000434* 012703 MOV R3,(R3)+ ;LOAD THE TRANSMIT CSR INTO LINKING TABLE
242 000436* 062702 ADD #2,R2 ;UPDATE THE ADDRESS POINTER
243 000442* 000746 BR 15 ;GO SET UP NEXT DEVICE
244
245
```

```
246 000444* 012767 002646* 001734 SETUP2: MOV #XMTQUE,XMTQPI ;SET UP TRANSMIT QUE ENTRY(IN)
247 ;POINTER
248 000452* 012767 002646* 001730 MOV #XMTQUE,XMTQPO ;SET UP TRANSMIT QUE RETRIEVAL
249 ;(OUT) POINTER
250 000460* 012767 002666* 001724 MOV #RCVQUE,RCVQPI ;SET UP RECEIVER QUE POINTERS
251 000466* 012767 002666* 001720 MOV #RCVQUE,RCVQPO ;(OUT) POINTER
252 000474* 012767 002706* 001714 MOV #ERRQUE,ERRQPI ;SET UP ERROR QUE POINTERS
253 000476* 012767 002706* 001710 MOV #ERRQUE,ERRQPO ;
254 000510* 012704 000154 MOV #R0,R4 ;LOAD R4 WITH NUMBER OF BUFFER
255 ;LOCATIONS TO BE CLEARED
256 000514* 012703 002426* MOV #XMTBUF,R3 ;BEGIN CLEARING BUFFERS AT THE
257 ;TRANSMIT BUFFER
258 000520* 005023 1$: CLR (R3)+ ;ZERO EACH LOCATION AND POINT P3 TO
259 ;NEXT WORD
260 000522* 005304 DEC R4 ;HAVE ALL LOCATIONS BEEN CLEARED?
261 000524* 001375 BNE 15 ;IF NO, CLEAR THE NEXT ONE
262
263 000526* 016700 002224 ACTVATE: MOV SELECT,R0 ;LOAD DEVICE SELECTION PARAMETER
264 000532* 110067 002240 MOV R0,DOMFLG ;SETUP TRANSFER COMPLETION FLAG
265 000536* 016701 172248 MOV ADDR,R1 ;COPY THE BASE ADDRESS
266 000542* 012767 002446* 002220 MOV #RCVRF0,VA ;GET THE PHYSICAL ADDRESS OF FIRST
267 ;RECEIVER BUFFER
268 000550* 104415 000000* 002770* GETPA, BEGIN, VA ;GET PHYSICAL ADDRESS FROM 16-BIT VA
269 000556* 016767 002210 002174* MOV PA,RCVADR ;SAVE RECEIVE BUFFER PHYSICAL ADDRESS
270 000564* 012767 002422* 002176* MOV #SYNC,VA ;GET THE PHYSICAL ADDRESS OF TRANSMITTER BUFFER
271 000572* 104415 000000* 002770* GETPA, BEGIN, VA ;GET PHYSICAL ADDRESS FROM 16-BIT VA
272 000580* 016767 002166 002154* MOV PA,XMTADR ;SAVE TRANSMITTER BUFFER PHYSICAL ADDRESS
273 000586* 000241 CLC ;BE SURE CARRY BIT IS CLEAR BEFORE ROTATING BITS
274 000610* 006167 002160 ROL EA ;ALIGN THE EXTENDED ADDRESS BITS IN GRADE
275 ;TO SET BITS 13 AND 14 OF THE REG/ERR
276 ;REGISTER
277 000614* 006200 1$: ASR R0 ;ISOLATE DEVICE SELECTION FLAG IN "CM" BIT
278 000616* 103407 BCS 35 ;IF SELECTED, GO INITIATE TRANSFER
279 000620* 001536 B2Q INITIAL ;IF NO MORE SELECTED, GO SETUP DATA
280 000626* 062701 000010 002124 2$: ADD #20,RCVADR ;UPDATE POINTER TO NEXT DEVICE ADDRESS
281 000632* 062767 000020 002124 ADD #20,RCVADR ;UPDATE POINTER TO NEXT RECEIVER BUFFER
282 000634* 000767 BR 15 ;PROCESS NEXT DEVICE
283 000636* 105061 000005 3$: CLRB 5(R1) ;CLEAR SECONDARY REGISTERS POINTER
284 ;FROM REG/ERR REGISTER (BITS 4-11)
285 000642* 012702 000020 MOV #20,R2 ;SET COUNT TO 16 FOR SECONDARY REGISTER
286 ;CLEARING
287 000646* 052761 010000 000004 4$: BIS #BIT12,4(R1) ;ENABLE EXTENDED BITS WRITING (REG/ERR BIT 12)
288 000654* 142761 000140 000005 BICB #140,5(R1) ;CLEAR BITS 16 AND 17 OF THE
289 ;SECONDARY REGISTER BY CLEARING 13
290 ;AND 14 OF THE REG/ERR REGISTER
291 000662* 005061 000006 CLR 6(R1) ;CLEAR A SECONDARY REGISTER
292 000664* 105261 000005 INCB 5(R1) ;POINT TO NEXT SECONDARY REGISTER
293 000672* 005302 DEC R2 ;REDUCE COUNT. ARE ALL SIXTEEN DONE?
294 000674* 001364 BNE 45 ;IF NO, GO DO NEXT ONE
295 000676* 012702 000020 MOV #20,R2 ;SET UP COUNTER TO CLEAR CHAR. DETECT REGS.
296 000706* 132761 000017 000001 MOVBR BR1,1(R1) ;IF YES, CLEAR CHARACTER DETECT REGISTERS
297 000710* 132761 000017 000001 BITB #17,1(R1) ;IS -BB OPTION HERE?
298 000716* 001417 BEQ 75 ;NO DONT CLEAR IT, CLEARING IT
299 ;WIPES OUT HIGH ORDER BITS.
300 000720* 112761 000030 000005 5$: MOVBR #30,5(R1) ;SELECT SECONDARY REGISTER 10
301 000726* 005061 000006 CLR 6(R1)
```

```

302 000732* 112761 000034 000005      MOV#  #34,5(R1)      ;SELECT SECONDARY REG. 14
303 000740* 005061 000006                CLR      6(R1)
304 000744* 005303 000000                DEC      R2
305 000748* 005303 000000                BEQ      R2
306 000750* 105361 000001                DECB    1(R1)
307 000754* 003361 000000                BGT     5$
308 000756* 112761 000012 000005 7$:  MOV#  #12,5(R1)      ;SELECT THE NEXT PAIR OF CHARACTER DETECT REGIST
309 000764* 052761 000040 000006        BIS     #BIT5,6(R1)  ;HIGH BYTE OF RECEIVER CSR
310 000772* 112761 000020 000005        MOV#  #20,5(R1)      ;SELECT SECONDARY REGISTER 12
311 001000* 156761 001770 000005        BISB   EA,5(R1)      ;MISCELLANEOUS REGISTER
312 001004* 015761 001746 000006        MOV#  RCVADR,6(R1)   ;MASTER CLEAR THE D011 BY
313 001014* 105261 000005 000006        INCB   5(R1)         ;SETTING BITS OF THE MISC REGISTER
314 001020* 012761 177751 000006        MOV#  #-15,6(R1)     ;ENABLE EXTENDED MEMORY BITS
315 001024* 012761 000022 000005        MOV#  #22,5(R1)     ;AND SELECT THE RECEIVER BUSS
316 001034* 156761 001734 000005        BISB   EA,5(R1)     ;ADDRESS REGISTER (SEC. REG.0)
317 001042* 016761 001714 000006        MOV#  XMTADR,6(R1)   ;SET THE EA BITS
318 001050* 105261 000005 000006        INCB   5(R1)         ;LOAD ADDRESS OF RECEIVER BUFFER
319 001054* 012761 177755 000006        MOV#  #-19,6(R1)    ;SELECT THE RECEIVER BYTE COUNT REGISTER
320 001062* 112761 000031 000005        MOV#  #31,5(R1)     ;(SECONDARY REG. 1)
321 001070* 016761 001326 000006        MOV#  SYNC,6(R1)    ;SET COUNT FOR 15 CHARACTERS
322 001076* 105261 000005 000006        INCB   5(R1)         ;SELECT THE TRANSMITTER BUSS ADDRESS REGISTER
323 001102* 012761 004010 000006        MOV#  #4010,6(R1)   ;SET THE EA BITS
324 001110* 052711 000042                BIS     #42,(R1)    ;(SECONDARY REGISTER 2)
325 001114* 000642                BR      2$          ;LOAD ADDRESS OF TRANSMITTER BUFFER
326 001116* 012701 002426*                INITIAL:MOV #XMTBUF,R1 ;SELECT THE TRANSMITTER BYTE COUNT REGISTER
327 001122* 012702 000017                MOV#  #15,R2        ;(SECONDARY REGISTER 3)
328 001126* 122767 000026 001643 1$:  CMPB   #26,DATA    ;LOAD TRANSMISSION BYTE COUNT
329 001134* 001009 000000                BNE     3$          ;SELECT THE SYNC CHARACTER REGISTER
330 001136* 105267 001635                INCB   DATA        ;(SECONDARY REGISTER 1)
331 001142* 105267 001631 5$:  MOV#  DATA,(R1)+   ;LOAD TRANSMISSION BYTE COUNT
332 001146* 105267 001625                INCB   DATA        ;SELECT THE SYNC CHARACTER REGISTER
333 001152* 005302 001625                DEC     R2           ;LOAD THE SYNC REGISTER WITH
334 001154* 001664 001625                BNE     4$          ;TWO COPIES OF THE SYNC CHARACTER
335 001156* 016700 001574                MOV#  SELECT,R0     ;SELECT THE MISC REGISTER (REG. 12)
336 001162* 016701 176620                MOV#  ADDR,R1       ;ENABLE LDR (BIT 5) AND
337 001166* 003404 000010 2$:  RSR     R0           ;SELECT 9 BITS/CHARACTER 1000 IN BITS 9-11)
338 001172* 001410 000010 3$:  BEQ     TMRSET      ;AND STRIP SYNC (BIT 1)
339 001174* 062701 000010 3$:  ADD     #10,R1      ;ADJUST POINTERS FOR NEXT DEVICE
340 001200* 000772                BR      2$

```

```

358 001202* 005211 000001 000002 4$:  INC     (R1)
359 001204* 012761 000051 000002      MOV#  #51,2(R1)     ;START RECEIVER (BIT 0=GO BIT)
360 001212* 000770                BR      3$          ;ENABLE TRANSMITTER (BIT 0) AND ERROR INTER-
361 001214* 012767 000005 001544 TMRSET: MOV #5,TMRCNT ;RUP (BIT 3) AND TRANSMITTER INTER-
362 001222* 005004 000000 35$:  TIMER: CLR R4       ;RUP (BIT 5)
363 001224* 104407 000000*        BREAK$,BEGIN      ;GO ADJUST POINTER TO NEXT DEVICE
364 001224* 104407 000000*        BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR...
365 001234* 105767 001536        TSTB   DONFLG      ;THEN CONTINUE AT NEXT INSTRUCTION
366 001240* 001433                BEQ     FINISH     ;IF DONFLG IS CLEAR, EACH SELECTED DEVICE WAS
367 001242* 005304                DEC     R4         ;SUCCESSFUL
368 001244* 001367 001514        BNE     3$        ;IF SO PERFORM ENDPASS HOUSEKEEPING
369 001246* 005367 001516        DEC     TMRCNT     ;IF NOT, REDUCE COUNT AND BREAK AGAIN
370 001254* 001363 001516        MOV#  DONFLG,R3    ;BREAK IF COUNT NOT EXCEEDED
371 001254* 116703                BNE     4$        ;REDUCE TIMING FACTOR
372 001260* 040367 001472        BIC    R3,SELECT   ;BREAK AGAIN IF NO TIMEOUT
373 001264* 006003 1$:  ROR     R3         ;IF TIMEOUT OCCURRED, SAVE PRESENT FLAGS
374 001266* 103402 3$:  RCS     2$        ;IN R3
375 001270* 005204 4$:  INC     R4         ;USE R3 TO DEACTIVATE HUNG DEVICE
376 001272* 000774 001104 2$:  BR      R4,NUMBA1 ;BY CLEARING ACTIVE SELECTION FLAG FROM
377 001274* 010467 001104 2$:  MOV#  NUMBA1,R4    ;DEVICE SELECTION PARAMETER
378 001300* 104420 000000* 002404*        OTOAS$,BEGIN,NUMBA1,M1 ;DETERMINE WHICH LINE WAS
379 001306* 002367*                ;***** ;BAD FOR REPORTING PURPOSES
380 001310* 104403 000000* 002354*        MSGNS$,BEGIN,HUNG ;IF THIS IS THE LINE R4 CONTAINS CORRECT
381 001316* 005767 001434                TST     SELECT     ;LINE NUMBER...GO REPORT IT
382 001322* 001002                BNE     FINISH     ;IF NOT, INCREMENT R4 WHICH WAS INITIALLY
383 001324* 104410 000000*                ENDS$,BEGIN      ;0 FROM THE PREVIOUS LOOP
384 001330* 104413 000000*        FINISH: ENDTIS$,BEGIN ;SIGNAL END OF ITERATION.
385 001334* 000167 177104                JMP     SETUP2     ;MONITOR SHOULD TEST END OF PASS
386 001340* 010577 001042        XMTINT: MOV R5,#XMTQPI ;START NEXT TRANSFER
387 001340* 010577 001042        XMTINT: MOV R5,#XMTQPI ;LOAD THE OFFSET TO THE CSR INTO TRANSMITTER QUE

```

414 001344* 062767 000002 001034 ADD #2,XMTQPI ;UPDATE THE QUEUE ENTRY POINTER
415 001352* 022767 002666* 001026 CMP #XMTQOE+20,XMTQPI ;HAS THE QUEUE BOUNDARY BEEN EXCEEDED?
416 001360* 001003 BNE IS ;IF IT HAS NOT EXCEEDED THE BOUNDARY,
417 ;GO DEFER SERVICE TO LEVEL 0
418 001362* 012767 002646* 001016 MOV #XMTQOE,XMTQPI ;IF IT HAS EXCEEDED THE POINTER
419 001370* 012605 1\$: MOV (SP),R5 ;RESTORE THE PREVIOUS R5 VALUE FROM STACK
420 ;-----
421 001372* 000004 000000* 001400* BIRQS,BEGIN,XMTRSRV ; QUEUE UP TO CONTINUE AT XMTRSRV AND RTI
422 ;-----
423 001400* 017701 001004 XMTRSRV: MOV #XMTQOE,R1 ;FETCH THE OFFSET FROM THE QUEUE
424 001404* 062767 000002 000776 ADD #2,XMTQOE ;UPDATE THE QUEUE RETRIEVAL POINTER
425 001412* 022767 002666* 000770 CMP #XMTQOE+20,XMTQOE ;HAS THE QUEUE BOUNDARY BEEN EXCEEDED?
426 BNE IS ;IF NOT, CONTINUE PROCESSING
427 001422* 012767 002646* 000760 MOV #XMTQOE,XMTQOE ;IF YES, RESET POINTER TO BEGINNING OF QUEUE
428 001430* 011100 1\$: MOV (R1),R0 ;LOAD THE CSR ADDRESS INTO R0
429 001432* 005760 000002 TST 2(R0) ;IF THERE WAS AN ERROR, BIT 15 OF REG/ERR IS SET
430 001436* 002017 BGE 2\$;IF NO ERROR, PROCESS THE INTERRUPT
431 001440* 010067 176434 MOV R0,CSRA ;ON ERROR, LOAD THE DEVICE ADDRESS INTO CSRA
432 001444* 011067 176432 MOV (R0),ACSR ;LOAD THE CONTENTS INTO ACSR
433 001450* 016067 000002 176426 MOV 2(R0),ASTAT ;LOAD THE REGISTER/ERRCP REG. INTO STATC
434 001456* 005067 176424 CLR ERRTYP ;UNKNOWN ERROR
435 ;*****
436 001462* 104405 000000* 000000 ;RDERS,BEGIN,NULL ;ERROR FLAG SET-CONTENTS OF REG/ERR IN STATC
437 ;*****
438 001470* 005004 CLR R4 ;COMPENSATE FOR PRINT DELAY ON TIMER
439 001472* 105067 001300 CLR# DNFLG
440 001476* 042710 000001 BIT #BIT0,(R0) ;DISABLE TRANSMITTER
441 001502* 104400 000000* 2\$: EXITS,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
442 ;-----
443
444
445 001506* 010577 000700 RCVINT: MOV R5,RCVQPI ;ENTER OFFSET INTO RECEIVER QUEUE
446 001512* 062767 000002 000672 ADD #RCVQPI ;UPDATE THE RECEIVER ENTRY POINTER
447 001520* 022767 002706* 000664 CMP #RCVQOE+20,RCVQPI ;HAS THE QUEUE BOUNDARY BEEN EXCEEDED?
448 001526* 001003 BNE IS ;IF NOT, CONTINUE PROCESSING
449 001530* 012767 002666* 000654 1\$: MOV #RCVQOE,RCVQPI ;IF SO, POINT POINTER TO QUEUE BEGINNING
450 001536* 012605 1\$: MOV (SP),R5 ;RESTORE PREVIOUS VALUE OF R5
451 ;-----
452 001540* 000004 000000* 001546* BIRQS,BEGIN,RCVSRV ; QUEUE UP TO CONTINUE AT RCVSRV AND RTI
453 ;-----
454 001546* 017700 000642 RCVSRV: MOV #RCVQOE,R0 ;RETRIEVE OFFSET FROM RECEIVER QUEUE
455 001552* 062767 000002 000634 ADD #2,RCVQOE ;UPDATE THE QUEUE POINTER
456 001560* 022767 002706* 000626 CMP #RCVQOE+20,RCVQOE ;HAS THE QUEUE BOUNDARY BEEN EXCEEDED?
457 001566* 001003 BNE IS ;IF NOT, CONTINUE PROCESSING
458 001570* 012767 002666* 000616 1\$: MOV #RCVQOE,RCVQOE ;IF SO, POINT POINTER TO QUEUE BEGINNING
459 001576* 011001 1\$: MOV (R0),R1 ;PLACE CSR ADDRESS IN R1
460 001600* 005761 000004 TST 4(R1) ;WERE THERE ANY ERRORS(BIT 15 OF REG/ERR REG)?
461 001604* 002014 BGE 2\$;IF NOT, CONTINUE PROCESSING
462 001606* 010167 176266 MOV R0,CSRA ;IF SO, LOAD BASE ADDRESS INTO CSRA
463 001612* 011167 176264 MOV (R1),ACSR ;MOVE CONTENTS OF CSR TO ACSR
464 001616* 016167 000004 176260 MOV 4(R1),ASTAT ;LOAD REG/ERR REG. CONTENTS INTO STATC
465 001624* 005067 176256 CLR ERRTYP ;UNKNOWN ERROR
466 ;*****
467 001630* 104405 000000* 000000 ;RDERS,BEGIN,NULL ;DQ11 ERROR FLAG SET-REG/ERR IN STATC
468 ;*****
469 001636* 122711 000246 2\$: CMPB #246,(R1) ;

470 001642* 001412 BEQ 3\$
471 001644* 010167 176230 MOV R1,CSRA
472 001650* 011167 176226 MOV (R1),ACSR
473 001654* 012767 000017 176224 MOV #1,ERRTYP ;UNKNOWN RECEIVER ERROR
474 ;*****
475 001662* 104405 000000* 000000 ;RDERS,BEGIN,NULL ;RECEIVER STATUS ERROR
476 ;*****
477 001670* 122761 000254 000002 3\$: CMPB #54,(R1) ;
478 001676* 001416 BNE 2\$;IF NO ERRORS,BEGIN COMPARING DATA
479 001700* 016167 000002 176174 MOV 2(R1),ACSR ;LOAD TRANSMITTER STATUS INTO ACSR
480 001706* 010167 176166 MOV R1,CSRA ;LOAD STATUS REGISTER ADDRESS
481 001712* 062767 000002 176160 ADD #2,CSRA ;ALTF# BASE ADDRESS TO SHOW TRANSMITTER ADDRESS
482 001720* 012767 000020 176160 MOV #2,ERRTYP ;UNKNOWN TRANSMITTER ERROR
483 ;*****
484 001726* 104405 000000* 000000 ;RDERS,BEGIN,NULL ;TRANSMITTER STATUS ERROR
485 ;*****
486
487
488 001734* 012767 002446* 001016 CKDATA: MOV #RCVQOE,RCVADR ;RESTORE BASE VALUE OF RECEIVER BUFFERS
489 001742* 012767 000017 001014 MOV #15,DATACT ;LOAD THE NUMBER OF CHARACTERS TO BE CHECKED
490 001750* 016002 000002 MOV 2(R0),R2 ;LOAD THE LINE NUMBER OF THIS DQ11
491 001754* 001405 1\$: BEQ 2\$;WHEN R2 IS 0, THE CORRECT RECEIVER BUFFER
492 ;HAS BEEN SELECTED
493 001756* 062767 000020 000774 ADD #20,RCVADR ;IF R2 IS NOT 0, POINT RCVADR TO NEXT BUFFER
494 001764* 005302 DEC R2 ;REDUCE THE LINE NUMBER BY 1
495 001766* 000772 BR 1\$;GO SEE IF THE CORRECT BUFFER HAS BEEN DETECTED
496 001770* 016702 000764 2\$: MOV RCVADR,R2 ;LOAD R2 WITH THE START OF THE RECEIVER BUFFER
497 001774* 012767 002446* 000756 2\$: MOV #RCVQOE,RCVADR ;RESTORE BASE VALUE OF RECEIVER BUFFERS
498 002002* 012703 002426* 3\$: MOV #XMTBUF,R3 ;LOAD R3 WITH THE START OF TRANSMITTER TEXT
499 002006* 122223 CMPB (R2),R3 ;MATCH CHARACTERS, ARE THEY THE SAME?
500 002010* 001453 BEQ 4\$;IF YES, GO PROCESS NEXT CHARACTER
501 002012* 010167 176062 MOV R1,CSRA ;LOAD THE BASE ADDRESS INTO CSRA
502 002016* 011167 176060 MOV (R1),ACSR ;LOAD RECEIVER CSR CONTENTS INTO ACSR
503 002022* 114267 176062 MOVB -R2,AWAS ;LOAD THE ACTUAL RECEIVED CHARACTER
504 002026* 114367 176054 MOVB -R3,ASB ;LOAD THE TRANSMITTED CHARACTER
505 002032* 010267 176046 MOV R2,WASADR ;LOAD THE CHARACTER RECEIVED ADDRESS
506 002036* 010367 176040 MOV R3,SADR ;LOAD THE CHARACTER TRANSMITTED ADDRESS
507 002042* 016705 000350 MOV ERRCPI,R5 ;USING R5, LOAD THE PRESENT VALUES OF THE REGISTER
508 ;IMAG THE ERROR QUEUE, UNQUEUEING THEM AFTER THE
509 ;DATA ERROR CALL, THIS ASSURES THAT THE REGISTER
510 ;CONTENTS ARE VALID
511 002046* 010025 MOV R0,(R5)+ ;SAVE THE QUEUE OFFSET
512 002050* 010125 MOV R1,(R5)+ ;SAVE THE CSR ADDRESS
513 002052* 010225 MOV R2,(R5)+ ;SAVE THE PRESENT RECEIVE BUFFER POINTER
514 002054* 010325 MOV R3,(R5)+ ;SAVE THE CURRENT TRANSMIT BUFFER POINTER
515 002056* 010425 MOV R4,(R5)+ ;SAVE THE TIMER COUNTDOWN
516 002060* 020527 002732* 5\$: MOV #ERRQOE+20, ;HAS THE POINTER EXCEEDED THE QUEUE BOUNDARY?
517 002064* 103402 BLD 5\$;IF SO, DO NOT RESET POINTER
518 002066* 012705 002706* MOV #ERRQOE,R5 ;POINT POINTER TO BEGINNING OF QUEUE
519 002072* 010567 000320 5\$: MOV #5,ERRQPI ;RESTORE ERROR QUE ENTRY POINTER
520 ;*****
521 002076* 104404 000000* ;DATERS,BEGIN ;DATA ERROR!!!
522 ;*****
523 002102* 016705 000312 MOV ERRCPI,R5 ;LOAD R5 WITH ERROR QUEUE RETRIEVAL POINTER
524 ;RETRIEVE, USING R5, THE FOLLOWING FROM THE ERRCP

```
526      002106* 012500      MOV      (R5)+,R0      ;QUEUE,PLACING THEM IN THE CORRESPONDING REGISTE
527      002110* 012501      MOV      (R5)+,R1      ;THE OFFSET
528      002112* 012502      MOV      (R5)+,R2      ;THE BASE ADDRESS....
529      002114* 012503      MOV      (R5)+,R3      ;THE CURRENT RECEIVER BUFFER POINTER....
530      002116* 012504      MOV      (R5)+,R4      ;THE CURRENT TRANSMITTER BUFFER POINTER...
531      002120* 012507      CMP      R5,#ERRQUE+20. ;AND THE CORRECT TIMER VALUE
532      002124* 103402      BLD      66           ;HAS THE POINTER EXCEEDED THE QUEUE BOUNDARY?
533      002126* 012705      MOV      #ERRQUE,R5    ;IF NOT,DO NOT RESET IT
534      002132* 010567      MOV      R5,ERRQPB    ;IF SO, RESET THE POINTER TO QUEUE BEGINNING
535      002136* 122223      CMPB    (R2)+,(R3)+   ;RESTORE THE RETRIEVAL POINTER
536      002140* 005367      DEC     DACT         ;INCREMENT THE BUFFER POINTERS TO
537      002144* 003320      RGT     3$          ;POINT TO THE CORRECT CHARACTER
538      002146* 005003      CLR     R3          ;REDUCE CHARACTER COUNT,ARE ALL 15 DONE?
539      002150* 016002      RCVDON: CLR    R3      ;IF YES,SETUP R3 TO TURN OFF FLAG IN DONFLG
540      002154* 000261      SEC     2(R0),R2    ;FOR THIS LINE
541      002156* 006103      RDL     R3          ;LOAD THE LINE NUMBER INTO R2
542      002160* 002375      DEC     R2          ;USING THE CARRY BIT, CREATE A ONE-BIT MASK
543      002162* 002375      RCE     R3          ;POINT THE MASK TO THE NEXT LINE
544      002164* 140367      BICB   R3,DONFLG   ;REDUCE THE LINE NUMBER
545      002170* 104400      EXITS$,BEGIN       ;IF POSITIVE OR 0, GO SHIFT BIT AGAIN
546      002174* 004567      LNKTAB: JSR    R5,RCVINT ;IF NEGATIVE, THE MASK IS CORRECTLY ALIGNED-
547      002200* 000000      0          ;CLEAR THE DONE BIT FOR THIS LINE
548      002202* 000000      0          ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
549      002204* 004567      JSR    R5,XMTINT    ;LINK FOR RECEIVER 0
550      002210* 000000      0          ;LINK FOR TRANSMITTER 0
551      002212* 004567      JSR    R5,RCVINT    ;LINK FOR RECEIVER 1
552      002214* 000000      0          ;LINK FOR TRANSMITTER 1
553      002216* 000000      1          ;LINK FOR RECEIVER 2
554      002218* 004567      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 2
555      002220* 000000      0          ;LINK FOR RECEIVER 3
556      002222* 004567      JSR    R5,RCVINT    ;LINK FOR TRANSMITTER 3
557      002224* 000000      0          ;LINK FOR RECEIVER 4
558      002226* 000000      2          ;LINK FOR TRANSMITTER 4
559      002228* 004567      JSR    R5,XMTINT    ;LINK FOR RECEIVER 5
560      002230* 000000      0          ;LINK FOR TRANSMITTER 5
561      002232* 004567      JSR    R5,RCVINT    ;LINK FOR RECEIVER 6
562      002234* 000000      0          ;LINK FOR TRANSMITTER 6
563      002236* 000000      1          ;LINK FOR RECEIVER 7
564      002238* 004567      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 7
565      002240* 000000      0          ;LINK FOR RECEIVER 8
566      002242* 004567      JSR    R5,RCVINT    ;LINK FOR TRANSMITTER 8
567      002244* 000000      0          ;LINK FOR RECEIVER 9
568      002246* 004567      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 9
569      002248* 000000      0          ;LINK FOR RECEIVER 10
570      002250* 004567      JSR    R5,RCVINT    ;LINK FOR TRANSMITTER 10
571      002252* 000000      0          ;LINK FOR RECEIVER 11
572      002254* 004567      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 11
573      002256* 000000      0          ;LINK FOR RECEIVER 12
574      002258* 004567      JSR    R5,RCVINT    ;LINK FOR TRANSMITTER 12
575      002260* 000000      0          ;LINK FOR RECEIVER 13
576      002262* 004567      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 13
577      002264* 000000      0          ;LINK FOR RECEIVER 14
578      002266* 004567      JSR    R5,RCVINT    ;LINK FOR TRANSMITTER 14
579      002268* 000000      0          ;LINK FOR RECEIVER 15
580      002270* 004567      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 15
581      002272* 000000      5          ;LINK FOR RECEIVER 16
```

```
582      002312* 004567      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 5
583      002316* 000000      0          ;LINK FOR RECEIVER 6
584      002320* 004567      JSR    R5,RCVINT    ;LINK FOR TRANSMITTER 6
585      002324* 000000      0          ;LINK FOR RECEIVER 7
586      002328* 000000      0          ;LINK FOR TRANSMITTER 7
587      002332* 004567      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 8
588      002336* 000000      0          ;LINK FOR RECEIVER 9
589      002340* 004567      JSR    R5,RCVINT    ;LINK FOR TRANSMITTER 9
590      002344* 000000      0          ;LINK FOR RECEIVER 10
591      002348* 000007      JSR    R5,XMTINT    ;LINK FOR TRANSMITTER 10
592      002352* 000000      0          ;LINK FOR RECEIVER 11
593      002356* 000000      0          ;LINK FOR TRANSMITTER 11
594      002360* 002360*      HUNG:  MESSAG      ;LINK FOR RECEIVER 12
595      002364* 004567      0          ;LINK FOR TRANSMITTER 12
596      002368* 042504      MESSAG: .ASCII "DEVICE " ;LINK FOR RECEIVER 13
597      002372* 000000      0          ;LINK FOR TRANSMITTER 13
598      002376* 040          ;LINK FOR RECEIVER 14
599      002380* 000006      M1:    .BLKW 6      ;LINK FOR TRANSMITTER 14
600      002384* 000000      M2:    .ASCIZ " HUNG" ;LINK FOR RECEIVER 15
601      002400* 000          ;LINK FOR TRANSMITTER 15
602      002404* 002404*      .EVEN ;LINK FOR RECEIVER 16
603      002408* 000000      NUMBA1: .WORD 0      ;LINK FOR TRANSMITTER 16
604      002412* 000000      XMTQPI: .WORD OPEN   ;LINK FOR RECEIVER 17
605      002416* 000000      XMTQPO: .WORD OPEN   ;LINK FOR TRANSMITTER 17
606      002420* 000000      RCVQPI: .WORD OPEN   ;LINK FOR RECEIVER 18
607      002424* 000000      RCVQPO: .WORD OPEN   ;LINK FOR TRANSMITTER 18
608      002428* 000000      ERRQPI: .WORD OPEN   ;LINK FOR RECEIVER 19
609      002432* 000000      ERRQPO: .WORD OPEN   ;LINK FOR TRANSMITTER 19
610      002436* 000000      0          ;LINK FOR RECEIVER 20
611      002440* 026          SYNC:   .BYTE 26,26  ;LINK FOR TRANSMITTER 20
612      002444* 026          0          ;LINK FOR RECEIVER 21
613      002448* 000010      XMTBUF: .BLKW 8      ;TRANSMITTER CHARACTER BUFFER
614      002452* 000010      RCVBF0: .BLKW 8      ;DEVICE 0 RECEIVER BUFFER
615      002456* 000010      RCVBF1: .BLKW 8      ;DEVICE 1 RECEIVER BUFFER
616      002460* 000010      RCVBF2: .BLKW 8      ;DEVICE 2 RECEIVER BUFFER
617      002464* 000010      RCVBF3: .BLKW 8      ;DEVICE 3 RECEIVER BUFFER
618      002468* 000010      RCVBF4: .BLKW 8      ;DEVICE 4 RECEIVER BUFFER
619      002472* 000010      RCVBF5: .BLKW 8      ;DEVICE 5 RECEIVER BUFFER
620      002476* 000010      RCVBF6: .BLKW 8      ;DEVICE 6 RECEIVER BUFFER
621      002480* 000010      RCVBF7: .BLKW 8      ;DEVICE 7 RECEIVER BUFFER
622      002484* 000010      0          ;LINK FOR TRANSMITTER 21
623      002488* 000010      XMTQUE: .BLKW 8      ;TRANSMITTER SERVICE QUEUE
624      002492* 000010      RCVQUE: .BLKW 8      ;RECEIVER SERVICE QUEUE
625      002496* 000024      ERRQUE: .BLKW 20     ;ERROR SERVICE QUEUE
626      002500* 000000      SELECT: OPEN        ;DEVICE SELECTION PARAMETER
627      002504* 000000      RCVADR: OPEN        ;PHYSICAL ADDRESS OF RECEIVER BUFFERS
628      002508* 000000      XMTADR: OPEN        ;PHYSICAL ADDRESS OF TRANSMITTER BUFFER
629      002512* 000000      0          ;FOR ALL DEVICES
630      002516* 000000      DATACT: .WORD OPEN   ;COUNTER FOR NUMBER OF DATA ITEMS
631      002520* 000000      THRCNT: .WORD OPEN   ;LOCATION FOR WATCHDOG TIMER FACTOR
632      002524* 000000      VA:    .WORD OPEN   ;LOCATION OF VIRTUAL ADDRESS PARAMETER
633      002528* 000000      PA:    .WORD OPEN   ;LOCATION OF PHYSICAL ADDRESS PARAMETER
634      002532* 000000      0          ;LINK FOR TRANSMITTER 22
635      002536* 000000      0          ;LINK FOR RECEIVER 23
636      002540* 000000      0          ;LINK FOR TRANSMITTER 23
637      002544* 000000      0          ;LINK FOR RECEIVER 24
```


XDQAI0, XDQAI0/SQL/CRF:SYM=DDXCOM, XDQAI0
RUN-TIME: 1 2 .3 SECONDS
RUN-TIME RATIO: 15/4=3.7
CORE USED: 7K (13 PAGES)