

DQ11

DEVICE ROUTINE (MPG)
MD-11-DTDQA-B

EP-DTDQA-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

This microfiche card contains a grid of frames. The frames are arranged in approximately 12 rows and 3 columns. Each frame contains a small, high-contrast image of a document page, likely containing technical data or code. The text within the frames is too small to be legible. The card is otherwise blank.

801

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DTDQA-B
 PRODUCT NAME: DQ11 DEVICE ROUTINE FOR MPG
 DATE: APRIL 1976
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHOR: W. R. GREENE

COPYRIGHT (C) 1975, 1976
 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

%

11-10-76
 4:09 PM
 DEC
 11-10-76
 4:09 PM
 DEC

55-098-11

.SBTTL REVISION HISTORY

- ; APR 76 DTDQA-B RELEASE
- ; JAN 76 ADDED MEMORY MANAGEMENT SUPPORT
- ; AUG 75 DTDQA-A INITIAL RELEASE

54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109

.SBTTL STANDARD DEVICE ROUTINE TABLE
.TITLE MAINDEC-11-DTDQA-B DQ11 DEVICE ROUTINE FOR MPG
;REVISION B
;FILENAME OF "TDQABO.MPG" ON MPG/XXDP MEDIA
;MACY11: DTDQA?,DTDQA?/CRF:SYM/DOC=DTDQA?.P11
;LNKX11: DTDQA?.MPG/B:0+DTDQA?/E
;PAPER TAPE: PUNCH DTDQA?.MPG/FILE:ELEV

000000'

.CSECT DQ11
.DSABL GBL

;THE FOLLOWING TABLE IS IN THE STANDARDIZED FORMAT REQUIRED
;TO INTERFACE WITH MPG.

000000' 007016
000002' 000000

LOCZ: .WORD DVREND-
FLAGWD: .WORD 0

;DEVICE ROUT SIZE IN BYTES
;DEVICE ROUT FLAGWORD

100000
040000
020000
010000
004000
002000
001000
000004
000002

DRWAIT= 100000
MRPBSY= 40000
MRSBSY= 20000
RDPBSY= 10000
RDSBSY= 4000
RNOABT= 2000
WNOABT= 1000
CLWVCT= 4
CLRVCT= 2

; DEVICE ROUTINE WAIT FLAG
; WRITE PRIMARY BUSY
; WRITE SECONDARY BUSY
; READ PRIMARY BUSY
; READ SECONDARY BUSY
; READ-DONT ABORT FLAG
; WRITE-DONT ABORT FLAG
; CLEAR WRITE VECTOR FLAG
; CLEAR READ VECTOR FLAG

000004' 000000
000006' 000000
000010' 000000
000012' 000000
000014' 000000
000016' 000000
000020' 000001
000022' 000000
000024' 160220
000026' 000300
000030' 000240
000032' 000240
000034' 004432
000036' 004464
000040' 001330
000042' 001206
000044' 001212
000046' 000000
000050' 000000
000052' 000000
000054' 000000
000056' 000000
000060' 000000

.WORD 0
.WORD 0
.WORD 0
.WORD 0
.WORD 0
.WORD 0
SIZE: .WORD 1
ERR: .WORD 0
DREGAD: .WORD 160220
IVCTAD: .WORD 300
RBUSRQ: .WORD 240
WBUSRQ: .WORD 240
.WORD HSKEEP-
.WORD REPORT-
.WORD KILL-
.WORD DATAER-
.WORD TOUTER-
CIOBSY: .WORD 0
CUPGER: .WORD 0
ULIST: .WORD 0
CLIST: .WORD 0
BINASC: .WORD 0
BTASLZ: .WORD 0

;INTERFACE WORD # 1 (NOT USED)
;INTERFACE WORD # 2 (NOT USED)
;INTERFACE WORD # 3 (NOT USED)
;INTERFACE WORD # 4 (NOT USED)
;INTERFACE WORD # 5 (NOT USED)
;INTERFACE WORD # 6 (NOT USED)
;# OF BYTES TRANSFERRED / UNIMAP FLG
;ERROR ON LAST I/O INDICATOR
;FIRST DEVICE REGISTER ADR
;INTERRUPT VECTOR ADR
;INT PROC STATUS WORD (BR 5)
;INT PROC STATUS WORD (BR 5)
;HOUSEKEEPING ROUT REL ADR
;REPORT ROUT REL ADR
;KILL ROUT REL ADR
;DATA ERROR COUNTER REL ADR
;TIME OUT ERROR ROUT REL ADR
;I/O BUSY BRANCH ADR
;DEVICE ERROR BRANCH ADR
;USER MODE PRINT ROUTINE BRANCH ADR
;CMND MODE PRINT ROUTINE BRANCH ADR
;CONVERT BINARY TO ASCII ROUT BR ADR
;CONVERT BINARY TO DECIMAL ASCII BR ADR

166									
167	000306'	120	201	DVCMDS:	.BYTE	120,201			
168	000310'	001164			.WORD	READ-			
169	000312'	130	201		.BYTE	130,201			
170	000314'	001720			.WORD	WRITE-			
171	000316'	376	000		.BYTE	376,0			
172	000320'	002670			.WORD	CRESET-			
173	000322'	375	000		.BYTE	375,0			
174	000324'	002610			.WORD	NOWAIT-			
175	000326'	374	000		.BYTE	374,0			
176	000330'	002412			.WORD	WAIT-			
177	000332'	373	000		.BYTE	373,0			
178	000334'	004166			.WORD	REPORT-			
179	000336'	372	000		.BYTE	372,0			
180	000340'	004162			.WORD	REPORT-			
191	000342'	371	000		.BYTE	371,0			
182	000344'	003176			.WORD	SELREG-			
183	000346'	370	000		.BYTE	370,0			
184	000350'	003206			.WORD	SELSEQ-			
185	000352'	367	000		.BYTE	367,0			
186	000354'	003256			.WORD	NOIDLE-			
187	000356'	366	000		.BYTE	366,0			
188	000360'	003232			.WORD	IDLE-			
189	000362'	365	000		.BYTE	365,0			
190	000364'	002606			.WORD	FDUPLX-			
191	000366'	364	000		.BYTE	364,0			
192	000370'	002564			.WORD	HDUPLX-			
193	000372'	363	000		.BYTE	363,0			
194	000374'	002734			.WORD	CALL-			
195	000376'	362	000		.BYTE	362,0			
196	000400'	002766			.WORD	LISTEN-			
197	000402'	361	000		.BYTE	361,0			
198	000404'	002724			.WORD	ANSWER-			
199	000406'	360	000		.BYTE	360,0			
200	000410'	003006			.WORD	HANGUP-			
201	000412'	357	000		.BYTE	357,0			
202	000414'	003050			.WORD	SEND-			
203	000416'	356	000		.BYTE	356,0			
204	000420'	003102			.WORD	RECV-			
205	000422'	177777			.WORD	177777			
206									
207	000424'	051103	051505	052105	DVPKTE:	.ASCII	/CRESET/		
208	000432'	376	000		.BYTE	376,0			
209	000434'	047516	040527	052111	.ASCII	/NOWAIT/			
210	000442'	375	000		.BYTE	375,0			
211	000444'	020040	040527	052111	.ASCII	/WAIT/			
212	000452'	374	000		.BYTE	374,0			
213	000454'	052123	052101	051525	.ASCII	/STATUS/			
214	000462'	373	000		.BYTE	373,0			
215	000464'	047503	047125	051524	.ASCII	/COUNTS/			
216	000472'	372	000		.BYTE	372,0			
217	000474'	042523	051114	043505	.ASCII	/SELREG/			
218	000502'	371	000		.BYTE	371,0			
219	000504'	042523	051514	050505	.ASCII	/SELSEQ/			
220	000512'	370	000		.BYTE	370,0			
221	000514'	047516	042111	042514	.ASCII	/NOIDLE/			

```

;VALID DEVICE FUNCTIONS
;FLAG BYTE:
;BIT 7 = NPR DEV
;BIT 3 = MASSBUS DEV
;BIT 0 = 2 WORDS FOR ADR
;          (18 BIT ADRS)

```

;TABLE TERMINATOR

;PACK TABLE EXTENSION

222	000522'	367	000
223	000524'	020040	042111 042514
224	000532'	366	000
225	000534'	042106	050125 054114
226	000542'	365	000
227	000544'	042110	050125 054114
228	000552'	364	000
229	000554'	020040	040503 046114
230	000562'	363	000
231	000564'	044514	052123 047105
232	000572'	362	000
233	000574'	047101	053523 051105
234	000602'	361	000
235	000604'	040510	043516 050125
236	000612'	360	000
237	000614'	020040	042523 042116
238	000622'	357	000
239	000624'	020040	042522 053103
240	000632'	356	000
241			
242	000634'	000376	001122
243	000640'	000375	001122
244	000644'	000374	001122
245	000650'	000373	001122
246	000654'	000372	001122
247	000660'	000371	001123
248	000664'	000370	001123
249	000670'	000367	001122
250	000674'	000366	001122
251	000700'	000365	001122
252	000704'	000364	001122
253	000710'	000363	001122
254	000714'	000362	001122
255	000720'	000361	001122
256	000724'	000360	001122
257	000730'	000357	001122
258	000734'	000356	001122
259			
260			
261			
262			
263	000740'	003	376
264	000742'	004537	000012
265	000746'	003	375
266	000750'	004537	000012
267	000754'	003	374
268	000756'	004537	000012
269	000762'	004	373
270	000764'	004537	000012 001002
271	000772'	004	372
272	000774'	004537	000012 001001
273	001002'	004	371
274	001004'	004537	000012 000000
275	001012'	004	370
276	001014'	004537	000012 000000
277	001022'	003	367

.BYTE	367,0
.ASCII	/ IDLE/
.BYTE	366,0
.ASCII	/FDUPLX/
.BYTE	365,0
.ASCII	/HDUPLX/
.BYTE	364,0
.ASCII	/ CALL/
.BYTE	363,0
.ASCII	/LISTEN/
.BYTE	362,0
.ASCII	/ANSWER/
.BYTE	361,0
.ASCII	/HANGUP/
.BYTE	360,0
.ASCII	/ SEND/
.BYTE	357,0
.ASCII	/ RECV/
.BYTE	356,0

DVMVTE: .WORD 376,LCRST-LOCZ ;MODEL VECTOR TBL EXTEN.
 .WORD 375,LNWAIT-LOCZ
 .WORD 374,LWAIT-LOCZ
 .WORD 373,LSTATS-LOCZ
 .WORD 372,LCOUNT-LOCZ
 .WORD 371,LSLREG-LOCZ
 .WORD 370,LSLSEQ-LOCZ
 .WORD 367,LNIDLE-LOCZ
 .WORD 366,LIDLE-LOCZ
 .WORD 365,LFULL-LOCZ
 .WORD 364,LHALF-LOCZ
 .WORD 363,LCALL-LOCZ
 .WORD 362,LLISEN-LOCZ
 .WORD 361,LANSWR-LOCZ
 .WORD 360,LHNGUP-LOCZ
 .WORD 357,LEND-LOCZ
 .WORD 356,LRECV-LOCZ

COMPILER TABLE EXTENSION

⋮
 DVCPTTE: .BYTE 3,376 ;CONTROL RESET
 .WORD 4537,10.
 .BYTE 3,375 ;NO WAIT
 .WORD 4537,10.
 .BYTE 3,374 ;WAIT
 .WORD 4537,10.
 .BYTE 4,373 ;STATUS
 .WORD 4537,10.,1002
 .BYTE 4,372 ;COUNTS
 .WORD 4537,10.,1001
 .BYTE 4,371 ;SELECT REGISTER
 .WORD 4537,10.,0
 .BYTE 4,370 ;SELECT SEQUENCE
 .WORD 4537,10.,0
 .BYTE 3,367 ;NO IDLE

278	001024'	004537	000012	.WORD	4537, 10.	
279	001030'	003	366	.BYTE	3, 366	; IDLE
280	001032'	004537	000012	.WORD	4537, 10.	
281	001036'	003	365	.BYTE	3, 365	; FULL DUPLEX
282	001040'	004537	000012	.WORD	4537, 10.	
283	001044'	003	364	.BYTE	3, 364	; HALF DUPLEX
284	001046'	004537	000012	.WORD	4537, 10.	
285	001052'	003	363	.BYTE	3, 363	; CALL
286	001054'	004537	000012	.WORD	4537, 10.	
287	001060'	003	362	.BYTE	3, 362	; LISTEN
288	001062'	004537	000012	.WORD	4537, 10.	
289	001066'	003	361	.BYTE	3, 361	; ANSWER
290	001070'	004537	000012	.WORD	4537, 10.	
291	001074'	003	360	.BYTE	3, 360	; HANGUP
292	001076'	004537	000012	.WORD	4537, 10.	
293	001102'	003	357	.BYTE	3, 357	; SEND
294	001104'	004537	000012	.WORD	4537, 10.	
295	001110'	003	356	.BYTE	3, 356	; RECEIVE
296	001112'	004537	000012	.WORD	4537, 10.	
297	001116'	177777		.WORD	177777	; TABLE END

298				:		
299				:		
300				:	DEVICE INTERFACE WORD SYMBOL TABLE	
301				:		
302	001120'	177777		DVIWST:	.WORD 177777	
303				:		

304				:		
305				:	MODEL STATEMENT TABLE EXTENSION	
306				:		

307	001122'			LCRST:		
308	001122'			LNWAIT:		
309	001122'			LWAIT:		
310	001122'			LSTATS:		
311	001122'			LCOUNT:		
312	001122'			LNIDLE:		
313	001122'			LIDLE:		
314	001122'			LFULL:		
315	001122'			LHALF:		
316	001122'			LCALL:		
317	001122'			LLISEN:		
318	001122'			LANSWR:		
319	001122'			LHNGUP:		
320	001122'			LEND:		
321	001122'	000		LRECV:	.BYTE 0	
322	001123'			LSLREG:		
323	001123'	377	000	LSLSEQ:	.BYTE 377, 0	
324		001126'			.EVEN	
325						
326		001126'		MSKPST=	.	
327		001126'		ISTAT=	.	; STORAGE FOR DEV REG'S AT INT
328	001126'	000000		RCSR:	.WORD 0	
329	001130'	000000		TCSR:	.WORD 0	
330	001132'	000000		RERR:	.WORD 0	
331	001134'	000000		MISC:	.WORD 0	
332						
333	001136'	000024		CSTAT:	.BLKW 20.	; DEV REG CURRENT VALUE STORAGE


```

334
335 001206' 000000          BYRD:  .WORD  0          ;BYTES READ COUNT
336 001210' 000000          .WORD  0
337 001212' 000000          BYWR:  .WORD  0          ;BYTES WRITTEN COUNT
338 001214' 000000          .WORD  0
339 001216' 000000          RDCNT: .WORD  0          ;READ CMND COUNT
340 001220' 000000          WRCNT: .WORD  0          ;WRITE CMND COUNT
341 001222' 000000          MISCNT: .WORD  0         ;MISC. CMND COUNT
342 001224' 000000          RICNT: .WORD  0         ;READ INTERRUPT COUNT
343 001226' 000000          WICNT: .WORD  0         ;WRITE INTERRUPT COUNT
344 001230' 000000          VRCNT: .WORD  0         ;VRC ERROR COUNT
345 001232' 000000          BCCNT: .WORD  0         ;BCC ERROR COUNT
346 001234' 000000          RNEMCT: .WORD  0        ;RCV NEM ERROR COUNT
347 001236' 000000          TNEMCT: .WORD  0        ;XMIT NEM ERROR CNT
348 001240' 000000          RLATCT: .WORD  0        ;RCV LATENCY ERROR CNT
349 001242' 000000          TLATCT: .WORD  0        ;XMIT LATENCY ERROR CNT
350 001244' 000000          RCLKCT: .WORD  0        ;RCV CLOCK LOSS COUNT
351 001246' 000000          TCLKCT: .WORD  0        ;XMIT CLOCK LOSS COUNT
352 001250' 000000          DATAER: .WORD  0      ;DATA ERROR COUNT
353 001252' 000000          TOECNT: .WORD  0      ;# OF ENTRIES INTO T/O ERROR ROUT
354
355 001254' 000000          FLAG:  .WORD  0          ;FLAGWORD STORAGE
356
357
358          001256'          HSKPEN= .
359
360          000000          XXXX=  0          ;VALUE TO BE TAILORED BY DEV ROUT
361

```

```

363                                     .SBTTL DQ11 FUNCTION ROUTINES
364
365                                     ;TIMEOUT ERROR HANDLER
366
367 001256' 005267 177770 TOUTER: INC TOECNT ; INCR TIME OUT ERROR COUNT
368 001262' 026727 177764 000010 CMP TOECNT, #8. ; EXCEEDED 8 TIMEOUTS?
369 001270' 001401 BEQ 2$ ; YES - CONTINUE
370 001272' 000205 RTS R5 ; NO - RETURN
371 001274' 004067 004134 2$: JSR R0, SAVREG ; SAVE REGISTERS
372 001300' 004767 004162 JSR PC, SUPTAD ; P TBL ADR TO R3
373 001304' 042713 000010 BIC #W4IOT, (R3) ; RESET WAIT FOR I/O TERM
374 001310' 005004 CLR R4
375 001312' 004567 004554 JSR R5, PRINT ; PRINT TIMEOUT ERR MSG
376 001316' 000026 .WORD TOEMSG-
377 001320' 000023 .WORD 19.
378 001322' 004567 000042 JSR R5, KILL ; KILL THE PROGRAM
379 001326' 004767 003624 JSR PC, ERDIRG ; DISPLAY STATUS & STMT #
380 001332' 004067 004112 JSR R0, RESREG ; RESTORE REGISTERS
381 001336' 012605 TOUTEX: MOV (SP)+, R5 ; GO DISPLAY DEVICE REGS
382 001340' 000177 176504 JMP @CUPGER
383
384 001344' 050504 030461 052040 TOEMSG: .ASCII 'DQ11 TIMEOUT ON I/O'
    001352' 046511 047505 052125
    001360' 047440 020116 027511
    001366' 117
    001370'
385                                     .EVEN
386
387                                     ;KILL USER PROGRAM ROUTINE
388
389 001370' 016702 176430 KILL: MOV DREGAD, R2 ; GET DEV REG ADR
390 001374' 032712 000040 BIT #40, (R2) ; READ INT UP?
391 001400' 001005 BNE DIDIST ; YES - CHECK FURTHER
392 001402' 032762 000040 000002 BIT #40, 2(R2) ; WRITE INT UP?
393 001410' 001001 BNE DIDIST ; YES - CHECK FURTHER
394 001412' 000425 BR KILLEX ; NO - EXIT
395 001414' 004567 004566 DIDIST: JSR R5, TRVECT ; TEST READ INT VECTOR
396 001420' 000407 BR CWRINT ; BRANCH IF NOT ME
397 001422' 042712 000041 BIC #41, (R2) ; RESET INT EBL AND GO
398 001426' 042767 014000 176346 BIC #RDPBSY+RDSBSY, FLAGWD ; CLEAR BUSY FLAGS
399 001434' 004767 002764 JSR PC, RPRINTV ; RESET INT VECTOR INFO
400 001440' 004567 004572 CWRINT: JSR R5, TWVECT ; TEST WRITE INT VECTOR
401 001444' 000410 BR KILLEX ; BRANCH IF NOT ME
402 001446' 042762 000041 000002 BIC #41, 2(R2) ; RESET INT EBL AND GO
403 001454' 042767 060000 176320 BIC #WRPBSY+WRSBSY, FLAGWD ; CLEAR BUSY FLAGS
404 001462' 004767 002754 JSR PC, RWINTV ; RESET INT VECTOR INFO
405 001466' 005067 176330 KILLEX: CLR ERR ; CLEAR ERROR INDICATOR
406 001472' 000205 RTS R5 ; RETURN
407
408                                     ;READ COMMAND HANDLER
409
410 001474' 010567 003630 READ: MOV R5, STMT ; SAVE R5
411 001500' 162767 000004 003622 SUB #4, STMT ; FOR STMT # REFERENCE
412 001506' 016704 176312 MOV DREGAD, R4 ; DEV REG ADDR TO R4
413 001512' 052767 002000 176262 BIS #RNOABT, FLAGWD ; SET READ - NO ABORT FLAG
414 001520' 032714 000040 BIT #40, (R4) ; TEST REC INT EBL
415 001524' 001407 BEQ SRDVEC ; BRANCH IF DQ NOT IN USE

```

K01

MAINDEC-11-DTDQA-B
DTDQAB.P11

DQ11 DEVICE ROUTINE FOR MPG
DQ11 FUNCTION ROUTINES

MACY11 27(732) 24-SEP-76 14:09 PAGE 4-1

SEQ 0072

416	001526'	032767	014000	176246		BIT	#RDPBSY+RDSBSY,FLAGWD	; THIS PGM USING IT?
417	001534'	001036				BNE	TMAINT	; YES, CONTINUE
418	001536'	004577	176304			JSR	R5, JCIOSBY	; ELSE RELEASE CONTROL
419	001542'	000754				BR	READ	
420	001544'	016767	176256	000012	SRDVEC:	MOV	IVCTAD, 10\$; INT VECTOR ADDR TO R1
421	001552'	016767	176252	000006		MOV	RBUSRQ, 20\$; ALSO BUS PRIORITY
422	001560'	004577	176302			JSR	R5, JSETVEC	; GO SET THE VECTOR
423	001564'	000000			10\$:	.WORD	XXXX	
424	001566'	000000			20\$:	.WORD	XXXX	
425	001570'	002414				.WORD	RDINT-	
426	001572'	016767	176230	000020	WVCLR:	MOV	IVCTAD, 10\$; INT VECTOR ADDR TO CALL
427	001600'	062767	000004	000012		ADD	#4, 10\$; ADJUST FOR WRITE INT
428	001606'	016767	176220	000006		MOV	WBUSRQ, 20\$; ALSO PASS BUS PRIORITY
429	001614'	004577	176246			JSR	R5, JSETVEC	; GO SET THE VECTOR
430	001620'	000000			10\$:	.WORD	XXXX	
431	001622'	000000			20\$:	.WORD	XXXX	
432	001624'	002042				.WORD	WRINT-	
433	001626'	005067	177420			CLR	TOECNT	; CLEAR TIME OUT ERR CNT
434	001632'	004767	003630		TMAINT:	JSR	PC, SUPTAD	; GET P TBL BASE IN R3
435	001636'	005067	176160			CLR	ERR	; CLEAR ERROR INDICATOR
436	001642'	005063	000030			CLR	PTOCNT(R3)	; CLEAR TIMEOUT COUNTER
437	001646'	005267	177344			INC	RDCNT	; INCR READ CMD COUNT
438	001652'	012764	005000	000004		MOV	#5000, 4(R4)	; SELECT MISC REG
439	001660'	032763	000200	000002		BIT	#SOPER, PPSW(R3)	; TEST MAINT BIT IN OPSW
440	001666'	001404				BEQ	RDMNCL	
441	001670'	052764	000010	000006		BIS	#10, 6(R4)	; SET TEST LOOP TO SAME STATE
442	001676'	000403				BR	RDMNST	
443	001700'	042764	000010	000006	RDMNCL:	BIC	#10, 6(R4)	
444	001706'	012500			RDMNST:	MOV	(R5)+, R0	; GET MSB OF ADDRESS
445	001710'	000241				CLC		
446	001712'	006000				ROR	R0	
447	001714'	006000				ROR	R0	
448	001716'	006000				ROR	R0	
449	001720'	006000				ROR	R0	
450	001722'	010067	001724			MOV	R0, MSADDR	; SAVE IT
451	001726'	032767	014000	176046		BIT	#RDPBSY+RDSBSY,FLAGWD	; IS READ PRI OR SEC SET?
452	001734'	001426				BEQ	RDPRIM	; NO - READ PRIMARY
453	001736'	032767	100000	176036		BIT	#DRWAIT, FLAGWD	; YES-TEST NOWAIT FLAG
454	001744'	001003				BNE	RDNOWT	; BRANCH IF SET
455	001746'	004577	176074			JSR	R5, JCIOSBY	; ELSE RELEASE CONTROL
456	001752'	000755				BR	RDMNST	
457	001754'	032767	010000	176020	RDNOWT:	BIT	#RDPBSY, FLAGWD	; READ PRIMARY BUSY?
458	001762'	001044				BNE	RDSEC	; YES-GO READ SEC
459	001764'	032767	004000	176010		BIT	#RDSBSY, FLAGWD	; READ SEC BUSY?
460	001772'	001407				BEQ	RDPRIM	; NO-GO READ PRIMARY
461	001774'	004577	176046		RPSBSY:	JSR	R5, JCIOSBY	; YES-RELEASE CONTROL
462	002000'	032767	002000	175774		BIT	#RNOABT, FLAGWD	; TEST READ-NO ABORT FLAG
463	002006'	001362				BNE	RDNOWT	; KEEP TRYING IF SET
464	002010'	000500				BR	RABORT	; ELSE GET OUT
465	002012'	032767	010000	175762	RDPRIM:	BIT	#RDPBSY, FLAGWD	; IS PRIMARY BUSY?
466	002020'	001365				BNE	RPSBSY	; YES - LET GO
467	002022'	005064	000004			CLR	4(R4)	; SELECT RBAP
468	002026'	056764	001620	000004		BIS	MSADDR, 4(R4)	; INSERT MSB'S
469	002034'	012564	000006			MOV	(R5)+, 6(R4)	; GET ADDRESS
470	002040'	012764	000400	000004		MOV	#400, 4(R4)	; SELECT RCCP
471	002046'	011564	000006			MOV	(R5), 6(R4)	; GET BYTE COUNT

```

472 002052' 012567 001602      MOV      (R5)+,RDPBC      ;SAVE IT
473 002056' 005464 000006      NEG      6(R4)          ;NEGATE IT
474 002062' 005725              TST      (R5)+          ;ADJUST R5
475 002064' 052767 010000 175710  BIS      #RDPBSY,FLAGWD ;SET READ PRIM BSY
476 002072' 000431              BR       RSETUP
477 002074' 032767 004000 175700  RDSEC:  BIT      #RDSBSY,FLAGWD ;IS SECONDARY BUSY?
478 002102' 001334              BNE     RPSBSY         ;YES - LET GO
479 002104' 012764 002000 000004  MOV      #2000,4(R4)    ;SELECT RBAS
480 002112' 056764 001534 000004  BIS      MSADDR,4(R4)   ;INSERT MSB'S
481 002120' 012564 000006      MOV      (R5)+,6(R4)   ;GET ADDRESS
482 002124' 012764 002400 000004  MOV      #2400,4(R4)   ;SELECT RCCS
483 002132' 011564 000006      MOV      (R5),6(R4)    ;GET BYTE COUNT
484 002136' 012567 001520      MOV      (R5)+,RDSBC   ;SAVE IT
485 002142' 005464 000006      NEG      6(R4)          ;NEGATE IT
486 002146' 005725              TST      (R5)+          ;ADJUST R5
487 002150' 052767 004000 175624  BIS      #RDSBSY,FLAGWD ;SET READ SEC BSY
488 002156' 052763 000010 000000  RSETUP: BIS      #WT4IOT,PFLGWD(R3) ;SET WAIT FOR IOT
489 002164' 052714 000041              BIS      #41,(R4)      ;SET GO AND INT EBL
490 002170' 052764 000010 000002  BIS      #10,2(R4)     ;SET ERR INT ENABLE
491 002176' 032767 100000 175576  BIT      #DRAIT,FLAGWD ;TEST THE NOWAIT FLAG
492 002204' 001004              BNE     RDDNWT         ;IF SET, DONT WAIT
493 002206' 000167 000552              JMP     TST'EB        ;WAIT UNTIL ALL I/O DONE
494 002212' 062705 000006      RABORT: ADD      #6,R5    ;ADJUST R5
495 002216' 042763 000010 000000  RDDNWT: BIC      #WT4IOT,PFLGWD(R3) ;CLEAR WAIT FOR IOT
496 002224' 042767 002000 175550  BIC      #RNOABT,FLAGWD ;CLEAR READ-NO ABORT FLAG
497 002232' 000205              RTS                    ;RETURN
498
499
500                                ;WRITE COMMAND HANDLER
501 002234' 010567 003070      WRITE:  MOV      R5,STMT   ;SAVE R5
502 002240' 162767 000004 003062  SUB      #4,STMT       ;FOR STMT # REFERENCE
503 002246' 016704 175552              MOV      DRGAD,R4
504 002252' 052767 001000 175522  BIS      #WNOABT,FLAGWD ;SET WRITE-NO ABORT FLAG
505 002260' 032764 000040 000002  BIT      #40,2(R4)     ;TEST WRITE INT EBL
506 002266' 001407              BEQ     SWRVEC         ;BRANCH IF DO NOT IN USE
507 002270' 032767 060000 175504  BIT      #WRPBSY+WRSBSY,FLAGWD ;THIS PGM USING IT?
508 002276' 001023              BNE     TLOOP
509 002300' 004577 175542              JSR     R5,JCIOBSY    ;YES-CONTINUE
510 002304' 000753              BR       WRITE        ;ELSE RELEASE CONTROL
511 002306' 016767 175514 000020  SWRVEC: MOV      IVCTAD,10$ ;INT VECTOR TO R1
512 002314' 062767 000004 000012  ADD      #4,10$        ;ADJUST FOR WRITE INT
513 002322' 016767 175504 000006  MOV      #WBUSRQ,20$   ;ALSO PASS BUS PRIORITY
514 002330' 004577 175532              JSR     R5,ASETVEC   ;GO SET THE VECTOR
515 002334' 000000      10$:  .WORD   XXXX
516 002336' 000000      20$:  .WORD   XXXX
517 002340' 001326              .WORD   WRINT-
518 002342' 005067 176704              CLR     TOECNT        ;CLEAR TIME OUT ERR CNT
519 002346' 004767 003114      TLOOP: JSR     PC,SUPTAD ;GET P TBL BASE IN R3
520 002352' 005067 175444              CLR     ERR           ;CLEAR ERROR INDICATOR
521 002356' 005063 000030              CLR     PTOCNT(R3)   ;CLEAR TIMEOUT COUNTER
522 002362' 005267 176632              INC     WRCNT         ;INCR WRITE CMD CNT
523 002366' 012764 005000 000004  MOV      #5000,4(R4)   ;SELECT MISC REG
524 002374' 032763 000200 000002  BIT      #SOPER,POPSW(R3) ;TEST MAINT BIT IN OPSW
525 002402' 001404              BEQ     WRMNCL
526 002404' 052764 000010 000006  BIS      #10,6(R4)    ;SET TEST LOOP BIT TO SAME STATE
527 002412' 000403              BR       WRMNST

```

```

528 002414' 042764 000010 000006 WRMNCL: BIC      #10,6(R4)
529 002422' 012500          WRMNST: MOV      (R5)+,R0          ;GET MSB OF ADDRESS
530 002424' 000241          CLC
531 002426' 006000          ROR      R0
532 002430' 006000          ROR      R0
533 002432' 006000          ROR      R0
534 002434' 006000          ROR      R0
535 002436' 010067 001210  MOV      R0,MSADDR          ;SAVE IT
536 002442' 032767 060000 175332 BIT      #WRPBSY+WRSBSY,FLAGWD ;IS WRITE PRI OR SEC SET?
537 002450' 001426          BEQ      WRPRIM          ;NO-WRITE PRIMARY
538 002452' 032767 100000 175322 BIT      #DRWAIT,FLAGWD    ;YES-TEST NOWAIT FLAG
539 002460' 001003          BNE      WRNOWT          ;BRANCH IF SET
540 002462' 004577 175360  JSR      R5,JCIOBSY      ;ELSE RELEASE CONTROL
541 002466' 000755          BR       WRMNST
542 002470' 032767 040000 175304 WRNOWT: BIT      #WRPBSY,FLAGWD    ;WRITE PRIMARY BUSY?
543 002476' 001045          BNE      WRSEC          ;YES-GO WRITE SEC
544 002500' 032767 020000 175274 BIT      #WRSBSY,FLAGWD    ;WRITE SEC BUSY
545 002506' 001407          BEQ      WRPRIM          ;NO-GO WRITE PRIMARY
546 002510' 004577 175332  WPSBSY: JSR      R5,JCIOBSY      ;YES-RELEASE CONTROL
547 002514' 032767 001000 175260 BIT      #WNOABT,FLAGWD    ;TEST WRITE-NO ABORT FLAG
548 002522' 001362          BNE      WRNOWT          ;KEEP TRYING IF SET
549 002524' 000476          BR       WABORT          ;ELSE GET OUT
550 002526' 032767 040000 175246 WRPRIM: BIT      #WRPBSY,FLAGWD    ;IS PRIMARY BUSY?
551 002534' 001365          BNE      WPSBSY          ;YES - LET GO
552 002536' 012764 001000 000004 MOV      #1000,4(R4)      ;SELECT WBAP
553 002544' 056764 001102 000004 BIS      MSADDR,4(R4)    ;INSERT MSB'S
554 002552' 012564 000006          MOV      (R5)+,6(R4)    ;GET ADDRESS
555 002556' 012764 001400 000004 MOV      #1400,4(R4)    ;SELECT WCCP
556 002564' 011564 000006          MOV      (R5),6(R4)    ;GET BYTE COUNT
557 002570' 012567 001060          MOV      (R5)+,WRPBC    ;SAVE IT
558 002574' 005464 000006          NEG      6(R4)          ;NEGATE IT
559 002600' 005725          TST     (R5)+          ;ADJUST R5
560 002602' 052767 040000 175172 BIS      #WRPBSY,FLAGWD    ;SET WRITE PRIM BSY
561 002610' 000431          BR       WSETUP
562 002612' 032767 020000 175162 WRSEC: BIT      #WRSBSY,FLAGWD    ;IS SECONDARY BUSY?
563 002620' 001333          BNE      WPSBSY          ;YES - LET GO
564 002622' 012764 003000 000004 MOV      #3000,4(R4)    ;SELECT WBAS
565 002630' 056764 001016 000004 BIS      MSADDR,4(R4)    ;INSERT MSB'S
566 002636' 012564 000006          MOV      (R5)+,6(R4)    ;GET ADDRESS
567 002642' 012764 003400 000004 MOV      #3400,4(R4)    ;SELECT WCCS
568 002650' 011564 000006          MOV      (R5),6(R4)    ;GET BYTE COUNT
569 002654' 012567 000776          MOV      (R5)+,WRSBC    ;SAVE IT
570 002660' 005464 000006          NEG      6(R4)          ;NEGATE IT
571 002664' 005725          TST     (R5)+          ;ADJUST R5
572 002666' 052767 020000 175106 BIS      #WRSBSY,FLAGWD    ;SET WRITE SEC BSY
573 002674' 052713 000010  WSETUP: BIS      #WT4IOT,(R3)    ;SET WAIT FOR IOT
574 002700' 052764 000051 000002 BIS      #51,2(R4)      ;SET GO AND INT EBL(DONE&ERR)
575 002706' 032767 100000 175066 BIT      #DRWAIT,FLAGWD    ;TEST THE NOWAIT FLAG
576 002714' 001004          BNE      WRDNWT          ;IF SET, DONT WAIT
577 002716' 000167 000042          JMP      TSTIEB          ;WAIT UNTIL ALL I/O DONE
578 002722' 062705 000006          WABORT: ADD     #6,R5      ;ADJUST R5
579 002726' 042713 000010  WRDNWT: BIC      #WT4IOT,(R3)    ;CLEAR WAIT FOR IOT
580 002732' 042767 001000 175042 BIC      #WNOABT,FLAGWD    ;CLEAR WRITE-NO ABORT FLAG
581 002740' 000205          RTS      R5            ;RETURN
582
583
;WAIT COMMAND HANDLER

```

```

584
585 002742' 042767 100000 175032 WAIT: BIC #DRWAIT,FLAGWD ;CLEAR DEV ROUT NOWAIT
586 002750' 016704 175050 MOV DREGAD,R4 ;POINT R4 AT REG ADDR
587 002754' 004767 002506 JSR PC,SUPTAD
588 002760' 052713 000010 BIS #WT4IOT,(R3) ;SET WAIT FOR IOT
589 002764' 032714 000040 TSTIEB: BIT #40,(R4) ;TEST RECV INT
590 002770' 001055 BNE RELEAS ;IF SET, RELEASE CONTROL
591 002772' 032764 000040 000002 BIT #40,2(R4) ;TEST XMIT INT
592 003000' 001051 BNE RELEAS ;IF SET, RELEASE CONTROL
593 003002' 032767 000002 174772 TRMTST: BIT #CLRVCT,FLAGWD ;TEST IF VECTOR CLR REQD
594 003010' 001410 BEQ 10$ ;BRANCH IF NOT
595 003012' 004567 003170 JSR R5,TRVECT ;TEST READ VECTOR
596 003016' 000405 BR 10$ ;BRANCH IF NOT ME
597 003020' 004767 001400 JSR PC,RRINTV ;GO RESET THE VECTOR
598 003024' 042767 000002 174750 BIC #CLRVCT,FLAGWD ;CLEAR THE REQ FLAG
599 003032' 032767 000004 174742 10$: BIT #CLWVCT,FLAGWD ;TEST IF VECTOR CLR REQD
600 003040' 001410 BEQ ERRTST ;BRANCH IF NOT
601 003042' 004567 003170 JSR R5,TWVECT ;TEST WRITE VECTOR
602 003046' 000405 BR ERRTST ;BRANCH IF NOT ME
603 003050' 004767 001366 JSR PC,RWINTV ;GO RESET THE VECTOR
604 003054' 042767 000004 174720 BIC #CLWVCT,FLAGWD ;CLEAR THE REQ FLAG
605 003062' 005767 000576 ERRTST: TST ERRFLG ;TEST FOR ANY ERROR
606 003066' 001421 BEQ RETN ;RETURN IF NONE
607 003070' 012767 000001 174724 MOV #1,ERR ;SET ERROR INDICATOR
608 003076' 032763 020000 000002 BIT #PRONER,POPSW(R3) ;TEST DONT PRINT ON ERROR BIT
609 003104' 001005 BNE ERREXT ;EXIT IF SET
610 003106' 005004 CLR R4
611 003110' 005267 001672 INC ABBREV ;SET ABBREVIATED RPT FLG
612 003114' 000167 001734 JMP ERRRPT ;GO REPORT ERRORS
613 003120' 000177 174724 ERREXT: JMP @CUPGER
614 003124' 004577 174716 RELEAS: JSR R5,@CIOBSY
615 003130' 000715 BR TSTIEB
616 003132' 000205 RETN: RTS ;RETURN INLINE
617
618 ;NO WAIT COMMAND HANDLER
619
620
621 003134' 052767 100000 174640 NOWAIT: BIS #DRWAIT,FLAGWD ;SET NOWAIT FLAG
622 003142' 004767 002320 JSR PC,SUPTAD
623 003146' 042713 000010 BIC #WT4IOT,(R3) ;CLEAR WAIT FOR IOT
624 003152' 000767 BR RETN
625
626 ;HALF DUPLEX COMMAND HANDLER
627
628 003154' 016704 174644 HDUPLX: MOV DREGAD,R4
629 003160' 052714 000010 BIS #10,(R4) ;SET HALF DUPLEX BIT
630 003164' 005267 176032 INC MISCNT ;INCR MISC CMD COUNT
631 003170' 000205 RTS R5 ;RETURN
632
633 ;FULL DUPLEX COMMAND HANDLER
634
635 003172' 016704 174626 FDUPLX: MOV DREGAD,R4
636 003176' 042714 000010 BIC #10,(R4) ;CLEAR HALF DUPLEX BIT
637 003202' 005267 176014 INC MISCNT ;INCR MISC CMD COUNT
638 003206' 000205 RTS R5 ;RETURN
639

```

```

640                                     ;CRESET COMMAND HANDLER
641
642 003210' 016704 174610      CRESET: MOV      DREGAD,R4      ;DEV REG ADDR TO R4
643 003214' 012764 005000 070004  MOV      #5000,4(R4)    ;SELECT MISC REG
644 003222' 012764 000040 000006  MOV      #40,6(R4)    ;SET MASTER CLEAR
645 003230' 012700 000010      MOV      #10,R0      ;SET UP COUNT
646 003234' 005001      CLR      R1
647 003236' 010164 000004 1$:      MOV      R1,4(R4)    ;SELECT SEC REGISTER
648 003242' 005064 000006      CLR      6(R4)      ;CLEAR IT
649 003246' 062701 000400      ADD      #400,R1    ;SET UP FOR NEXT REG
650 003252' 005300      DEC      R0        ;DONE ?
651 003254' 001370      BNE     1$        ;NO-CLEAR NEXT ONE
652 003256' 012700 000020      MOV      #16.,R0
653 003262' 005014      CLR      (R4)      ;SELECT 1ST SEQ REG
654 003264' 012764 006000 000004  MOV      #6000,4(R4) ;SELECT SEQ AS SECND REG
655 003272' 005064 000006 2$:      CLR      6(R4)      ;CLEAR IT
656 003276' 062714 000400      ADD      #400,(R4)  ;SELECT NEXT ONE
657 003302' 005300      DEC      R0
658 003304' 001372      BNE     2$        ;REPEAT FOR ALL
659 003306' 012764 005000 000004  MOV      #5000,4(R4) ;SELECT MISC REG
660 003314' 012764 004000 000006  MOV      #4000,6(R4) ;SET BITS/CHAR= 8
661 003322' 005067 174474      CLR      ERR      ;CLEAR ERROR INDICATOR
662 003326' 000205      RTS      R5        ;RETURN
663
664                                     ;CALL AND ANSWER COMMAND HANDLER
665
666 003330'
667 003330' 016704 174470      CALL:  ANSWER: MOV     DREGAD,R4      ;DEV REG ADDR TO R4
668 003334' 052764 001000 000002  BIS     #1000,2(R4)  ;SET DTR
669 003342' 032764 002000 000002  BIT     #2000,2(R4) ;TEST DSR
670 003350' 001003      BNE     CALLOK     ;EXIT WHEN READY
671 003352' 004577 174470      JSR     R5,DCIOBSY ;ELSE RELEASE CONTROL
672 003356' 000764      BR      CALL
673 003360' 005267 175636      CALLOK: INC     MISCNT
674 003364' 000205      RTS     R5        ;RETURN
675
676                                     ;LISTEN COMMAND HANDLER
677
678 003366' 016704 174432      LISTEN: MOV     DREGAD,R4      ;DEV REG ADDR TO R4
679 003372' 032764 004000 000002  BIT     #4000,2(R4) ;TEST RING IND
680 003400' 001003      BNE     RANG      ;EXIT WHEN DETECTED
681 003402' 004577 174440      JSR     R5,DCIOBSY ;ELSE RELEASE CONTROL
682 003406' 000767      BR      LISTEN
683 003410' 005267 175606      RANG:  INC     MISCNT
684 003414' 000205      RTS     R5        ;RETURN INLINE
685
686                                     ;HANGUP COMMAND HANDLER
687
688 003416' 016704 174402      HANGUP: MOV     DREGAD,R4
689 003422' 042764 000400 000002  DISCON: BIC     #400,2(R4) ;LOWER RTS
690 003430' 012700 000017      MOV     #15.,R0
691 003434' 012701 000553      HNGDL1: MOV     #363.,R1
692 003440' 005301      HNGDL2: DEC     R1        ;DELAY 15 MS
693 003442' 001376      BNE     HNGDL2
694 003444' 005300      DEC     R0
695 003446' 001372      BNE     HNGDL1

```

```

696 003450' 042764 001000 000002      BIC      #1000,2(R4)      ;LOWER DTR
697 003456' 005267 175540              INC      MISCNT
698 003462' 000205              RTS      RS              ;RETURN INLINE
699
700                                ;SEND COMMAND HANDLER
701
702 003464' 016704 174334      SEND:    MOV      DREGAD,R4
703 003470' 052764 000400 000002      BIS      #400,2(R4)      ;RAISE RTS
704 003476' 032764 020000 000002      BIT      #20000,2(R4)   ;TEST CTS
705 003504' 001003              BNE      SENDOK         ;EXIT WHEN SET
706 003506' 004577 174334      JSR      RS,SCI0BSY     ;ELSE RELEASE CONTROL
707 003512' 000764              BR       SEND
708 003514' 005267 175502      SENDOK: INC      MISCNT
709 003520' 000205              RTS      RS              ;RETURN INLINE
710
711                                ;RECEIVE COMMAND HANDLER
712
713 003522' 016704 174276      RECV:   MOV      DREGAD,R4
714 003526' 042764 000400 000002      BIC      #400,2(R4)      ;LOWER RTS
715 003534' 005267 175462              INC      MISCNT
716 003540' 000205              RTS      RS              ;RETURN INLINE
717
718                                ;SELECT REGISTER COMMAND HANDLER
719
720 003542' 016704 174256      SELREG: MOV      DREGAD,R4
721 003546' 012500              MOV      (RS)+,R0      ;GET DESIRED REG #
722 003550' 110064 000005      MOVSB   R0,5(R4)      ;SELECT IT
723 003554' 000205              RTS      RS              ;RETURN INLINE
724
725                                ;SELECT SEQUENCE COMMAND HANDLER
726
727 003556' 016704 174242      SELSEQ: MOV      DREGAD,R4
728 003562' 012500              MOV      (RS)+,R0      ;GET DESIRED SEQ/CHAR ADR
729 003564' 020027 000020      CMP     R0,#20        ;IF LESS THAN OCTAL 20
730 003570' 103401              BLO     1$            ;PASS ON VALUE
731 003572' 011000              MOV     (R0),R0       ;ELSE GET CONTENTS
732 003574' 042700 177760      1$:    BIC     #177760,R0   ;ZERO ALL BUT 4 LSB'S
733 003600' 042714 007400      BIC     #007400,(R4)  ;CLEAR CHAR BITS IN RCSR
734 003604' 150064 000001      BISB   R0,1(R4)      ;SET DESIRED BITS
735 003610' 000205              RTS      RS              ;RETURN INLINE
736
737                                ;IDLE COMMAND HANDLER
738
739 003612' 016704 174206      IDLE:   MOV      DREGAD,R4
740 003616' 052764 000002 000002      BIS     #2,2(R4)      ;SET IDLE MODE
741 003624' 005267 175372              INC     MISCNT
742 003630' 000205              RTS      RS              ;RETURN INLINE
743
744                                ;NO IDLE COMMAND HANDLER
745
746 003632' 016704 174166      NOIDLE:MOV      DREGAD,R4
747 003636' 042764 000002 000002      BIC     #2,2(R4)      ;RESET IDLE MODE
748 003644' 005267 175352              INC     MISCNT
749 003650' 000205              RTS      RS              ;RETURN INLINE
750

```


;WRITE & READ CONTROL TABLES

752
753
754
755
756
757
758
759
760
761

003652' 000000
003654' 000000
003656' 000000
003660' 000000
003662' 000000
003664' 000000

MSADDR: .WORD 0
WRPBC: .WORD 0
WRSBC: .WORD 0
RDPBC: .WORD 0
RDSBC: .WORD 0
ERRFLG: .WORD 0

:MOST SIGNIF ADR BITS
:INITIAL ADDRESSES

:ERROR INFO

```

763
764                                     ;WRITE INTERRUPT HANDLER
765
766
767 003666' 004067 001542          WRINT: JSR      R0, SAVREG      ;SAVE REGISTERS
768 003672' 004567 001610          JSR      R5, S1STAT    ;STORE DEVICE REG CONTENTS
769 003676' 175230                .WORD    ISTAT-.
770 003700' 005267 175322          INC      WICNT         ;INCR WRITE INT COUNT
771 003704' 004767 001556          JSR      PC, SUPTAD    ;SET INTERNAL PTRS
772 003710' 005764 000004          TST     4(R4)         ;TEST FOR ERROR
773 003714' 100477                BMI      ERRINT       ;DETOUR IF TRUE
774 003716' 032764 000200 000002  BIT     @200, 2(R4)    ;TEST PRIMARY DONE BIT
775 003724' 001025                BNE     WPDONE        ;BRANCH IF SET
776 003726' 042764 000100 000002  BIC     @100, 2(R4)    ;CLEAR SEC DONE
777 003734' 042767 020000 174040  BIC     @WRSBSY, FLAGWD ;CLEAR SEC BUSY FLAG
778 003742' 012764 003000 000004  MOV     @3000, 4(R4)   ;SELECT WBS
779 003750' 016700 177702          MOV     WRSBC, R0     ;GET WRITE SEC BC
780 003754' 010701                MOV     PC, R1        ;SET POINTER
781 003756' 062701 175236          ADD     @BYWR+2-., R1 ;BYTES WRITTEN
782 003762' 060011                ADD     R0, (R1)      ;THEN UPDATE IT
783 003764' 005541                ADC     -(R1)
784 003766' 032767 040000 174006  BIT     @WRPBSY, FLAGWD ;TEST PRIMARY BUSY FLAG
785 003774' 001430                BEQ     WRTERM        ;IF NOT SET, STOP WRITES
786 003776' 000442                BR      INTEXT        ;ELSE WAIT FOR NEXT INT
787 004000' 042764 000200 000002  WPDONE: BIC     @200, 2(R4)  ;CLEAR PRIMARY DONE
788 004006' 042767 040000 173766  BIC     @WRPBSY, FLAGWD ;CLEAR PRI BSY FLG
789 004014' 012764 001000 000004  MOV     @1000, 4(R4)   ;SELECT WBP
790 004022' 016700 177626          MOV     WRPBC, R0     ;GET WRITE PRIM BC
791 004026' 010067 173766          MOV     R0, SIZE     ;UPDATE BYTES TRANSFERED
792 004032' 010701                MOV     PC, R1        ;SET POINTER TO
793 004034' 062701 175160          ADD     @BYWR+2-., R1 ;BYTES WRITTEN COUNT
794 004040' 060011                ADD     R0, (R1)      ;THEN UPDATE IT
795 004042' 005541                ADC     -(R1)
796 004044' 032767 020000 173730  BIT     @WRSBSY, FLAGWD ;TEST SEC BUSY FLAG
797 004052' 001401                BEQ     WRTERM        ;IF NOT SET, STOP WRITES
798 004054' 000413                BR      INTEXT        ;ELSE WAIT FOR NEXT INT
799 004056' 042764 000041 000002  WRTERM: BIC     @41, 2(R4) ;CLEAR GO AND INT EBL
800 004064' 032714 000040          BIT     @40, (R4)    ;TEST READ INT EBL
801 004070' 001005                BNE     INTEXT        ;DO NOTHING IF SET
802 004072' 042713 000010          BIC     @WT4IOT, (R3) ;ELSE RESET WAIT FOR IOT
803 004076' 052767 000004 173676  GVECAD: BIS     @CLWVCT, FLAGWD ;SET CLR VECT REQUEST
804 004104' 004067 001340          INTEXT: JSR     R0, RESREG ;RESTORE REGISTERS
805 004110' 000177 173760          JMP     @RTNINT       ;RETURN FROM INTERRUPT
806 004114' 116467 000004 177542  ERRINT: MOVB   4(R4), ERRFLG ;SAVE ERROR BITS
807 004122' 116700 177536          MOVB   ERRFLG, R0    ;GET ERROR BITS
808 004126' 010701                MOV     PC, R1
809 004130' 062701 175120          ADD     @DATAER-., R1 ;SET ERROR CNTR PNTR
810 004134' 012702 000010          MOV     @8., R2      ;SET ERROR BIT COUNT
811 004140' 006000                10$:   ROR     R0
812 004142' 103002                BCC    20$           ;TEST EACH BIT
813 004144' 005241                INC    -(R1)         ;IF SET INCR CTR
814 004146' 000401                BR     30$           ;
815 004150' 005741                20$:   TST    -(R1)         ;IF NOT - SKIP IT
816 004152' 005302                30$:   DEC    R2
817 004154' 001371                10$:   BNE    10$         ;REPEAT FOR ALL 8 BITS
818 004156' 012764 005000 000004  MOV     @5000, 4(R4)  ;SELECT MISC REG

```

819	004164'	052764	000040	000006
820	004172'	005067	173604	
821	004176'	042713	000010	
822	004202'	000500		

BIS	#40,6(R4)
CLR	FLAGWD
BIC	#WT4IOT,(R3)
BR	RERVEC

```

;SET MASTER CLEAR
;CLEAR PROG BSY BITS
;CLEAR WAIT FOR IOT
;GO RESTORE VECTORS AND EXIT

```

```

;READ INTERRUPT HANDLER
824
825
826
827 004204' 004067 001224      RDINT: JSR      RO, SAVREG      ;SAVE REGISTERS
828 004210' 004567 001272      JSR      RS, S1STAT      ;STORE DEVICE REG CONTENTS
829 004214' 174712      .WORD   ISTAT-
830 004216' 005267 175002      INC      RICNT           ;INCR READ INT COUNT
831 004222' 004767 001240      JSR      PC, SUPTAD      ;SET INTERNAL PTRS
832 004226' 032714 000200      BIT      #200, (R4)      ;TEST PRIMARY DONE BIT
833 004232' 001024      BNE      RPDONE         ;BRANCH IF SET
834 004234' 042714 000100      BIC      #100, (R4)      ;CLEAR SEC DONE
835 004240' 042767 004000 173534 BIC      #RDSBSY, FLAGWD ;CLEAR SEC BUSY FLAG
836 004246' 012764 002000 000004 MOV      #2000, 4(R4)    ;SELECT RBAS
837 004254' 016700 177402      MOV      RDSBC, RO      ;GET READ SEC BC
838 004260' 010701      MOV      PC, R1         ;SET POINTER
839 004262' 062701 174726      ADD      #BYRD+2--, R1  ;TO BYTES READ COUNT
840 004266' 060011      ADD      RO, (R1)       ;THEN UPDATE IT
841 004270' 005541      ADC      -(R1)
842 004272' 032767 010000 173502 BIT      #RDPBSY, FLAGWD ;TEST PRIMARY BSY FLG
843 004300' 001426      BEQ      RDTERM        ;IF NOT SET, STOP READS
844 004302' 000700      BR      INTEXT         ;ELSE WAIT FOR NEXT INT
845 004304' 042714 000200      RPDONE: BIC      #200, (R4)   ;CLEAR PRIMARY DONE
846 004310' 042767 010000 173464 BIC      #RDPBSY, FLAGWD ;CLEAR PRIM BSY FLG
847 004316' 005064 000004      CLR      4(R4)         ;SELECT RBAP
848 004322' 016700 177332      MOV      RDPBC, RO     ;GET READ PRIM BC
849 004326' 010067 173466      MOV      RO, SIZE      ;UPDATE BYTES TRANSFERED
850 004332' 010701      MOV      PC, R1        ;SET POINTER TO
851 004334' 062701 174654      ADD      #BYRD+2--, R1  ;BYTES READ COUNT
852 004340' 060011      ADD      RO, (R1)       ;THEN UPDATE IT
853 004342' 005541      ADC      -(R1)
854 004344' 032767 004000 173430 BIT      #RDSBSY, FLAGWD ;TEST SEC BUSY FLAG
855 004352' 001401      BEQ      RDTERM        ;IF NOT SET, STOP READS
856 004354' 000653      BR      INTEXT         ;ELSE WAIT FOR NXT INT
857 004356' 042714 000041      RDTERM: BIC      #41, (R4) ;CLEAR GO AND INT EBL
858 004362' 032764 000040 000002 BIT      #40, 2(R4)     ;TEST XMIT INT EBL
859 004370' 001245      BNE      INTEXT        ;DO NOTHING IF SET
860 004372' 042713 000010      BIC      #WT4IOT, (R3) ;ELSE RESET WAIT FOR IOT
861 004376' 042764 000010 000002 BIC      #10, 2(R4)     ;AND CLEAR ERR INT EBL
862 004404' 052767 000002 173370 RERVEC: BIS      #CLR VCT, FLAGWD ;SET CLR READ VCT REQ
863 004412' 052767 000004 173362 BIS      #CLW VCT, FLAGWD ;SET CLR WR VCT REQ
864 004420' 000167 177460      JMP      INTEXT        ;EXIT
865
866
867 004424' 016767 173376 000004 RRINTV: MOV      IVCTAD, 10$ ;RESET READ VECT SUBR
868 004432' 004577 173432      JSR      RS, 2CLRVEC
869 004436' 000000      10$: .WORD   XXXX
870 004440' 000207      RTS      PC
871
872
873 004442' 016767 173360 000012 RWINTV: MOV      IVCTAD, 10$ ;RESET WRITE VECT SUBR
874 004450' 062767 000004 000004 ADD      #4, 10$
875 004456' 004577 173406      JSR      RS, 2CLRVEC
876 004462' 000000      10$: .WORD   XXXX
877 004464' 000207      RTS      PC

```

H02

MAINDEC-11-DTDQA-B
DTDQAB.P11

DQ11 DEVICE ROUTINE FOR MPG
DQ11 SUPPORT ROUTINES

MACY11 27(732) 24-SEP-76 14:09 PAGE 5

SEQ 0082

```
879          .SBTTL  DQ11 SUPPORT ROUTINES
880
881
882          ;DEVICE ROUTINE HOUSEKEEPING
883
884          ;JSR    R5,HSKEEP      S/R CALL
885          ;.WORD  0 OR 1        0 = DO HSKP PER OPSW
886          ;                1 = UNCOND. DO HSKP
887          ;R2 = PROG'S OPSW
888          ;DESTROYS R0,R1
889
890 004466' 005725      HSKEEP: TST    (R5)+      ;UNCONDITIONALLY DO HSKP?
891 004470' 001003      BNE    10$      ;N,Y-10$
892 004472' 032702 000004 BIT    #HSKPEP,R2  ;OPSW SPECIFY EACH PASS HSKP?
893 004476' 001010      BNE    30$      ;Y,N-30$
894 004500' 010700      10$:  MOV    PC,R0      ;SET UP FIRST WD ADR
895 004502' 062700 174424 ADD    #HSKPST-.,R0
896 004506' 012701 000054 MOV    #HSKPEN-HSKPST/2,R1 ;SET UP # OF WORDS
897 004512' 005020      20$:  CLR    (R0)+      ;HSKP ALL NECESSARY AREAS
898 004514' 005301      DEC    R1
899 004516' 001375      BNE    20$
900 004520' 000205      30$:  RTS    R5          ;EXIT IN-LINE
901
902
903          ;DQ11 REPORT ROUTINE
904
905          ;JSR    R5,REPORT      S/R CALL
906          ;.WORD  FLGWD        FLAGWORD
907          ;                BIT 15 = CMND MODE CALL
908          ;                BIT 9  = PROG STMT CALL
909          ;                BIT 1  = DO STATUS REPORT
910          ;                BIT 0  = DO COUNTS REPORT
911
912 004522' 005067 000260 REPORT: CLR    ABBREV      ;CLR ABBREVIATED RPT FLAG
913 004526' 004067 000702 JSR    R0,SAVREG      ;SAVE REG'S R0 - R5
914 004532' 032715 177776 BIT    #177776,(R5)  ;DISPLAYING CNTS AT END OF
915 004536' 001012      BNE    8$          ;PROG PASS? (Y,N-8$)
916 004540' 010700      MOV    PC,R0      ;SET UP ADR OF CNTS
917 004542' 062700 174444 ADD    #BYRD-.,R0
918 004546' 012701 000022 MOV    #18.,R1
919 004552' 005720      2$:  TST    (R0)+      ;GET # OF CNT WORDS
920 004554' 001003      BNE    8$          ;THIS CNT WORD = 0?
921 004556' 005301      DEC    R1          ;Y,N-8$
922 004560' 001374      BNE    2$          ;DECR WORD CNT
923 004562' 000475      BR     DVREX      ;CK'ED ALL WORDS? (Y,N-2$)
924 004564' 004767 000676 8$:  JSR    PC,SUPTAD    ;GO TO EXIT -- ALL CNTS ARE 0'S
925 004570' 012504      MOV    (R5)+,R4    ;SET UP PROG TBL ADR IN R3
926 004572' 032704 000002 BIT    #2,R4        ;GET FLAGWORD
927 004576' 001403      BEQ    10$       ;GOING TO DO STATUS DISPLAY?
928 004600' 004567 000756 JSR    R5,STSTAT    ;Y,N-10$
929 004604' 174332      .WORD  CSTAT-      ;GO STORE STATUS REG'S
930 004606' 032704 000002 10$: BIT    #2,R4        ;DISPLAY DEV STATUS?
931 004612' 001416      BEQ    DISCNT     ;Y,N-DISCNT
932 004614' 004567 001252 JSR    R5,PRINT     ;ISSUE 'AT INT' MSG
933 004620' 001461      .WORD  ATMSG-
934 004622' 000014      .WORD  12.
```

935	004624'	004567	001054	JSR	RS,DISPIS	;GO DISPLAY STATUS AT LAST INT
936	004630'	174276		.WORD	ISTAT-	
937	004632'	004567	001234	JSR	RS,PRINT	;ISSUE 'CURRENTLY' MSG
938	004636'	001457		.WORD	CURMSG-	
939	004640'	000012		.WORD	IO.	
940	004642'	004567	001044	JSR	RS,DISPST	;GO DISPLAY CURRENT STATUS
941	004646'	174270		.WORD	CSTAT-	
942	004650'	032704	000001	DISCNT: BIT	#1,R4	;DISPLAY COUNTS?
943	004654'	001434		BEQ	RPTEND	;Y,N-RPTEND
944	004656'	012700	000022	MOV	#18,R0	;SET UP NUMBER OF WORDS
945	004662'	010701		MOV	PC,R1	;SET UP ADR OF CNTS
946	004664'	062701	174322	ADD	#BYRD-.,R1	
947	004670'	010702		MOV	PC,R2	;SET UP TBL ADR
948	004672'	062702	000116	ADD	#REPTBL-.,R2	
949	004676'	012267	000012	RPTLP: MOV	(R2)+,RPTBAS	;MOV MSG ADR TO S/R LINKAGE
950	004702'	004067	000526	JSR	RO,SAVREG	;SAVE ALL REG'S
951	004706'	011100		MOV	(R1),R0	;GET CURRENT COUNT
952	004710'	004577	173142	JSR	RS,ABINASC	;CONVERT IT TO ASCII
953	004714'	000000		RPTBAS: .WORD	XXXX	
954	004716'	004067	000526	JSR	RO,RESREG	;RESTORE REG'S
955	004722'	005721		TST	(R1)+	;POINT AT NXT CNT
956	004724'	005300		DEC	R0	;DONE ALL WORDS?
957	004726'	001363		BNE	RPTLP	;Y,N-RPTLP
958	004730'	005767	000052	TST	ABBREV	;TEST ABBREVIATED RPT FLG
959	004734'	001014		BNE	PRERLN	;BRANCH IF SET
960	004736'	004567	001130	JSR	RS,PRINT	;GO ISSUE COUNTS MSG
961	004742'	001444		.WORD	CNTSMG-	
962	004744'	000410		.WORD	CNTSEN-CNTSMG	
963	004746'	004567	001120	RPTEND: JSR	RS,PRINT	;ISSUE "END OF REPORT" MSG
964	004752'	001355		.WORD	RENDMG-	
965	004754'	177763		.WORD	-13.	
966	004756'	004067	000466	DVREX: JSR	RO,RESREG	;RESTORE REGISTERS
967	004762'	005725		TST	(R5)+	;SET UP RETURN POINT
968	004764'	000205		RTS	RS	;EXIT IN-LINE
969						
970	004766'	004567	001100	PRERLN: JSR	RS,PRINT	;PRINT ERROR MESSAGE
971	004772'	001612		.WORD	ERRMSG-	
972	004774'	000212		.WORD	CNTSEN-ERRMSG	
973	004776'	005067	176662	CLR	ERRFLG	;CLEAR ERROR FLAG
974	005002'	000167	176112	JMP	ERREXT	;TAKE ERROR EXIT
975						
976	005006'	000000		ABBREV: .WORD	0	
977						
978						
979	005010'	001506		REPTBL: .WORD	BCMRD-RPTBAS	
980	005012'	001514		.WORD	BCMRD+6-RPTBAS	
981	005014'	001530		.WORD	BCMWR-RPTBAS	
982	005016'	001536		.WORD	BCMWR+6-RPTBAS	
983	005020'	001563		.WORD	CMDCRD-RPTBAS	
984	005022'	001576		.WORD	CMDCWR-RPTBAS	
985	005024'	001613		.WORD	CMDCMS-RPTBAS	
986	005026'	001645		.WORD	RDINMS-RPTBAS	
987	005030'	001660		.WORD	WRINMS-RPTBAS	
988	005032'	001707		.WORD	ERCVRC-RPTBAS	
989	005034'	001724		.WORD	ERCBC-RPTBAS	
990	005036'	001741		.WORD	ERRNEM-RPTBAS	

991	005040'	001756	.WORD	ERTNEM-RPTBAS
992	005042'	001776	.WORD	ERRLAT-RPTBAS
993	005044'	002013	.WORD	ERTLAT-RPTBAS
994	005046'	002030	.WORD	ERRCLK-RPTBAS
995	005050'	002045	.WORD	ERTCLK-RPTBAS
996	005052'	002074	.WORD	ERCOTA-RPTBAS

```

998                                     ;DQ11 ERROR REPORT ROUTINE
999
1000 005054' 004767 000004  ERRRPT: JSR   PC,ERRDIS
1001 005060' 000177 172764      JMP   @CUPGER
1002 005064' 010701      ERRDIS: MOV   PC,R1                ;POINT R1 AT ERR MSG
1003 005066' 062701 000260      ADD   @MSGBF-,R1
1004 005072' 012767 000014 000054  MOV   @12,ERMBCT
1005 005100' 010700      MOV   PC,R0                ;POINT R0 AT ERR MSG TBL
1006 005102' 062700 000144      ADD   @ERCDTB-,R0
1007 005106' 105710      1$:   TSTB  (R0)
1008 005110' 001416      BEQ   ERTBEN              ;BRANCH IF R0 AT TBL END
1009 005112' 132067 176546      BITB  (R0)+,ERRFLG      ;TEST FOR PARTICULAR ERR
1010 005116' 001003      BNE   3$                 ;BRANCH IF FOUND
1011 005120' 062700 000005      2$:   ADD   @5,R0
1012 005124' 000770      BR    1$
1013 005126' 012702 000005      3$:   MOV   @5,R2
1014 005132' 112021      4$:   MOVB  (R0)+,(R1)+    ;MOVE MSG CODE TO ERR MSG
1015 005134' 005267 000014      INC   ERMBCT            ;BUMP BYTE COUNT
1016 005140' 005302      DEC   R2
1017 005142' 001373      BNE   4$
1018 005144' 000760      BR    1$                ;CHECK IF MORE
1019 005146' 004567 000720      ERTBEN: JSR  R5,PRINT     ;PRINT ERROR MSG
1020 005152' 000160      .WORD EMSGHD-
1021 005154' 000014      ERMBCT: .WORD 12
1022 005156' 004567 000522      ERDIRG: JSR  R5,DISPIS   ;DISPLAY DEVICE REGS
1023 005162' 173744      .WORD ISTAT-
1024 005164' 016300 000022      ERRSNM: MOV   PSRCST(R3),R0 ;GET ADDR OF SRC STMTS
1025 005170' 111001      10$:  MOVB  (R0),R1        ;SAVE STMT LENGTH
1026 005172' 026067 000004 000130  CMP   4(R0),STMT        ;ERROR OCCUR ON THIS STMT?
1027 005200' 001402      BEQ   20$              ;YES - BRANCH
1028 005202' 060100      ADD   R1,R0            ;POINT AT NEXT STATEMENT
1029 005204' 000771      BR    10$             ;GO CK NEXT STMT
1030 005206' 005720      20$:  TST   (R0)+          ;SET UP ADR OF STMT # DATA
1031 005210' 010701      MOV   PC,R1            ;SET UP DATA OUTPUT ADDR
1032 005212' 062701 000214      ADD   @STMNUM-,R1
1033 005216' 004577 172640      JSR   R5,@DECASC       ;CONVERT IT TO ASCII
1034 005222' 012767 020040 000202  MOV   @20040,STMNUM+4  ;SET 2 LOW DIGITS TO SPACES
1035 005230' 004567 000636      JSR   R5,PRINT         ;ISSUE STMT # MSG
1036 005234' 000162      .WORD STNMNG-
1037 005236' 177762      .WORD -14
1038 005240' 005067 176420      CLR   ERRFLG          ;CLEAR ERROR FLAG
1039 005244' 000207      RTS   PC
1040
1041 005246' 020001 041524 045514  ERCDTB: .ASCII <001>/ TCLK/ ;ERROR MSG CODE TABLE
1042 005254' 020002 041522 045514      .ASCII <002>/ RCLK/
1043 005262' 020004 046124 052101      .ASCII <004>/ TLAT/
1044 005270' 020010 046122 052101      .ASCII <010>/ RLAT/
1045 005276' 020020 047124 046505      .ASCII <020>/ TNEM/
1046 005304' 020040 047122 046505      .ASCII <040>/ RNEM/
1047 005312' 020100 041122 041503      .ASCII <100>/ RBCC/
1048 005320' 020200 053122 041522      .ASCII <200>/ RVRC/
1049 005326' 000      .BYTE 0
1050 005330' 005330'      .EVEN
1051
1052 005330' 000000      STMT:  .WORD 0        ;SAVED R5 FOR STMT #
1053

```


L02

MAINDEC-11-DTDQA-B
DTDQAB.P11

DQ11 DEVICE ROUTINE FOR MPG
DQ11 SUPPORT ROUTINES

MACY11 27(732) 24-SEP-76 14:09 PAGE 5-4

SEQ 0086

1054	005332'	050504	030461	042440	EMSGHD: .ASCII	/DQ11 ERROR: /
	005340'	051122	051117	020072		
1055	005346'	000050			EMSGBF: .BLKB	40.
1056	005416'	052123	047115	020124	STMNMG: .ASCII	/STMNT # /
	005424'	020043				
1057	005426'	054130	054130	054130	STMNUM: .ASCII	/XXXXXX/

```

1059          .SBTTL  SUBROUTINES FOR DQ11 DEVICE ROUTINE
1060
1061          ;SAVE REGISTERS R0 THRU R5
1062
1063          ;JSR    R0,SAVREG          S/R CALL
1064
1065          SAVREG: MOV    R1,-(SP)          ;SAVE R0 THRU R5
1066          MOV    R2,-(SP)
1067          MOV    R3,-(SP)
1068          MOV    R4,-(SP)
1069          MOV    R5,-(SP)
1070          MOV    R0,PC          ;EXIT IN-LINE
1071
1072
1073          ;RESTORE REGISTERS R0 THRU R5
1074
1075          ;JSR    R0,RESREG          S/R CALL
1076
1077          RESREG: TST    (SP)+          ;RESTORE R4 THRU R0
1078          MOV    (SP)+,R5
1079          MOV    (SP)+,R4
1080          MOV    (SP)+,R3
1081          MOV    (SP)+,R2
1082          MOV    (SP)+,R1
1083          RTS    R0          ;EXIT IN-LINE
1084
1085
1086          ;SET PROGRAM'S PROG TABLE ADR IN R3
1087
1088          ;JSR    PC,SUPTAD          S/R CALL
1089
1090          SUPTAD: MOV    PC,R3          ;SET UP LOCATION ZERO ADR
1091          ADD    #LOCZ-,R3
1092          SUB    -2(R3),R3          ;SUBTRACT PROG TBL LENGTH
1093          MOV    DREGAD,R4          ;PUT DEV REG ADR IN R4
1094          RTS    PC          ;EXIT IN-LINE
1095
1096
1097          ;STORE DEVICE'S STATUS REGISTERS ON INTERRUPT
1098
1099          ;JSR    R5,SISTAT          S/R CALL
1100          ;.WORD  STADR-          REL STORAGE ADR
1101          ;DESTROYS R0,R1,R2
1102
1103          SISTAT: MOV    DREGAD,R1
1104          MOV    #12,5(R1)          ;SELECT MISC REG
1105          MOV    R5,R0          ;GET REL STORAGE ADR & MAKE
1106          ADD    (R5)+,R0          ;IT ABSOLUTE
1107          MOV    #DVREGE-DVREGS/6,-(SP) ;GET # OF REG'S TO STORE
1108          MOV    PC,R2          ;GET ADR OF 1ST REG DISPLACEMENT
1109          ADD    #DVREGS+4-.,R2
1110          MOV    (R2),R1          ;GET REG DISPLACEMENT
1111          ADD    DREGAD,R1          ;ADD IN REG'S BASE ADR
1112          MOV    (R1),(R0)+          ;STORE REGISTER VALUE
1113          ADD    #6,R2          ;POINT AT NXT DISPLACEMENT
1114          DEC    (SP)          ;DECR REG CNT

```

```

1115 005554' 001370      BNE      10$      ;DONE ALL? (Y,N-10$)
1116 005556' 005726      TST      (SP)+    ;CLEAN UP THE STACK
1117 005560' 000205      RTS       R5      ;EXIT IN-LINE
1118
1119
1120                               ;STORE DEVICE REGISTERS, INCLUDING SECONDARY
1121
1122                               ;JSR      R5,STSTAT      S/R CALL
1123                               ;WORD   STADR-      REL STORAGE ADR
1124                               ;DESTROYS RO,R1,R2
1125
1126 005562' 016701 172236 STSTAT: MOV      DREGAD,R1
1127 005566' 112761 000012 000005 MOVVB   #12,5(R1)      ;SELECT MISC REG
1128 005574' 010500      MOV      R5,R0      ;GET REAL STORAGE ADDRESS
1129 005576' 062500      ADD      (R5)+,R0    ;MAKE IT ABSOLOUTE
1130 005600' 012746 000004      MOV      #DVREGS-DVREGS/6,-(SP) ;GET # PRI REGS TO STORE
1131 005604' 010702      MOV      PC,R2      ;GET ADDR OF 1ST REG DISP
1132 005606' 062702 172314      ADD      #DVREGS+4-.,R2
1133 005612' 011201 10$:      MOV      (R2),R1    ;GET REG DISPLACEMENT
1134 005614' 066701 172204      ADD      DREGAD,R1  ;ADD IN REGS BASE ADR
1135 005620' 011120      MOV      (R1),(R0)+ ;STORE REGISTER VALUE
1136 005622' 062702 000006      ADD      #6,R2     ;POINT AT NEXT DISPLACEMENT
1137 005626' 005316      DEC      (SP)      ;DECR REG COUNT
1138 005630' 001370      BNE      10$      ;LOOP IF NOT DONE
1139 005632' 012716 000020      MOV      #16.,(SP) ;STORE SEC REG CNT
1140 005636' 016701 172162      MOV      DREGAD,R1
1141 005642' 011267 000034 20$:      MOV      (R2),SECREG ;GET SEC REG #
1142 005646' 042767 100000 000026 BIC     #100000,SECREG ;CLEAR MSB
1143 005654' 116761 000022 000005 MOVVB   SECREG,5(R1) ;SELECT IT
1144 005662' 016120 000006      MOV      6(R1),(R0)+ ;STORE ITS CONTENTS
1145 005666' 062702 000006      ADD      #6,R2     ;POINT AT NEXT REG #
1146 005672' 005316      DEC      (SP)      ;DECR REG CNT
1147 005674' 001362      BNE      20$      ;LOOP IF NOT DONE
1148 005676' 005726      TST      (SP)+    ;CLEAN UP THE STACK
1149 005700' 000205      RTS       R5      ;EXIT IN-LINE
1150 005702' 000000      SECREG: .WORD   0
1151
1152
1153                               ;TAILOR STATUS MSG & PRINT IT
1154
1155                               ;JSR      R5,DISPST      S/R CALL
1156                               ;WORD   STATADR-      REL ADR OF STATUS DATA
1157                               ;DESTROYS RO,R1,R2
1158
1159 005704' 012700 000004      DISPIS: MOV      #4,R0      ;SET UP COUNT FOR MAIN REGS
1160 005710' 000402      BR       TAILOR
1161 005712' 012700 000024      DISPST: MOV      #20.,R0    ;SET UP COUNT FOR ALL REGS
1162 005716' 010502      TAILOR: MOV      R5,R2    ;GET REL DATA ADR
1163 005720' 062502      ADD      (R5)+,R2    ;MAKE IT ABS
1164 005722' 010701      MOV      PC,R1      ;SET UP ADR OF REG NAMES IN ASCII
1165 005724' 062701 172172      ADD      #DVREGS-.,R1
1166 005730' 012167 000436 10$:      MOV      (R1)+,DVRGNG ;MOVE REG NAME TO MSG
1167 005734' 012167 000434      MOV      (R1)+,DVRGNG+2
1168 005740' 005721      TST      (R1)+    ;BYPASS DISP VALUE
1169 005742' 004067 177466      JSR      R0,SAVREG ;SAVE REG'S RO - R5
1170 005746' 011200      MOV      (R2),R0   ;GET REG'S STORED VALUE

```

```

1171 005750' 004577 172102      JSR      RS,JBINASC      ;CONVERT IT TO ASCII
1172 005754' 000424              .WORD    DVRGDT-
1173 005756' 004567 000110      JSR      RS,PRINT+      ;PRINT THE STATUS MSG
1174 005762' 000410              .WORD    DVRGMG-
1175 005764' 000014              .WORD    12.
1176 005766' 004067 177456      JSR      R0,RESREG      ;RESTORE R0 - R4
1177 005772' 005722              TST      (R2)+          ;POINT AT NXT REG VALUE
1178 005774' 005300              DEC      R0             ;DECR REG CNT
1179 005776' 001354              BNE      10$           ;DONE ALL? (Y,N-10$)
1180 006000' 000-                RTS      RS             ;EXIT IN-LINE
1181
1182                               ;DISPLAY CURRENT UNIT #
1183
1184                               ;JSR FC,DISUMM      S/R CALL
1185                               ;R3 MUST CONTAIN PROG TBL ADR
1186                               ;DESTROYS R0,R1,R2
1187
1188 006002' 012767 000022 000056 DISUMM: MOV      #18.,DISUML      ;INIT TO NORM MSG LNTH
1189 006010' 116300 000035          MOVB     PCURDV(R3),R0  ;GET CURRENT UNIT #
1190 006014' 020027 000020          CMP      R0,#16.      ;VALID UNIT #?
1191 006020' 101007              BHI      DISUIV        ;Y,N-DISUIV
1192 006022' 004577 172032      JSR      RS,JBATSLZ    ;CONVERT # TODECIMAL ASCII
1193 006026' 000336              .WORD    UNASCI-
1194 006030' 016767 000334 000326 MOV      UNASCI+4,UNASCI ;MOVE ASCII # TO 1ST TWO DIGITS
1195 006036' 000410              BR       DISUPR
1196 006040' 012767 000026 000020 DISUIV: MOV      #22.,DISUML ;SET UP ERR COND MSG LNTH
1197 006046' 042700 177400          BIC      #177400,R0   ;RESET HIGH BYTE
1198 006052' 004577 172000      JSR      RS,JBINASC    ;CONVERT BINARY TO ASCII
1199 006056' 000306              .WORD    UNASCI-
1200 006060' 004567 000006          DISUPR: JSR      RS,PRINT ;GO ISSUE UNIT # MSG
1201 006064' 000260              .WORD    UNITMG-
1202 006066' 000020          DISUML: .WORD    16.
1203 006070' 000207          RTS      PC           ;EXIT INLINE
1204

```

```

1206                                     ;ISSUE MSG TO LIST DEVICE
1207
1208                                     ;JSR   RS,PRINT          S/R CALL
1209                                     ;.WORD MSGADR-.        REL ADR OF MSG
1210                                     ;.WORD BYTCNT         MSG BYTE CNT (IF NEGATIVE,
1211                                     ;                       RESET PRT DEV DEDICATED.)
1212                                     ;R3 = PROG TBL ADR
1213                                     ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
1214                                     ;DESTROYS R0,R1,R2
1215
1216 006072' 010500          PRINT: MOV   RS,R0          ;GET MSG ADR & MAKE IT ABS
1217 006074' 062500          ADD   (RS)+,R0
1218 006076' 012501          MOV   (RS)+,R1          ;GET BYTE COUNT
1219 006100' 005704          TST   R4              ;USE CMND MODE PRINT?
1220 006102' 100030          BPL   40$            ;Y N-40$
1221 006104' 010702          MOV   PC,R2          ;SET UP LINK INFO ADR
1222 006106' 062702 000040  ADD   #20$--,R2
1223 006112' 160200          SUB   R2,R0          ;MAKE MSG ADR REL
1224 006114' 010022          MOV   R0,(R2)+      ;STORE MSG ADR
1225 006116' 010112          MOV   R1,(R2)       ;STORE MSG'S BYTE COUNT
1226 006120' 100001          BPL   10$            ;CNT NEG? (Y N-10$)
1227 006122' 005412          NEG   (R2)          ;MAKE IT POSITIVE
1228 006124' 016367 000006 000144 10$: MOV   PASCIN(R3),PROG#M ;STORE PROG'S # IN MSG
1229 006132' 004577 171716  JSR   RS,ACLIST      ;ISSUE PROG #
1230 006136' 000136          .WORD PNMMSG-.
1231 006140' 000005          .WORD 5
1232 006142' 004577 171706  JSR   RS,ACLIST      ;ISSUE MSG SPECIFIED
1233 006146' 000000 20$: .WORD XXXX
1234 006150' 000000          .WORD XXXX
1235 006152' 004577 171676  JSR   RS,ACLIST      ;ISSUE A <CR> & <LF>
1236 006156' 000302          .WORD CRLF-.
1237 006160' 000002          .WORD 2
1238 006162' 000410          BR    PRTEX          ;GO TO EXIT
1239 006164' 010067 000010 40$: MOV   R0,50$         ;STORE MSG'S ABS ADR
1240 006170' 010167 000006  MOV   R1,60$         ;STORE ITS BYTE CNT
1241 006174' 004577 171652  JSR   RS,ACLIST      ;GO TO MPG TO ISSUE THE MSG
1242 006200' 000000 50$: .WORD XXXX
1243 006202' 000000 60$: .WORD XXXX
1244 006204' 000205  PRTEX: RTS   R5          ;EXIT IN-LINE
1245
1246
1247                                     ;TEST READ INTERRUPT VECTOR S/R
1248
1249 006206' 016767 171614 000010 TRVECT: MOV   IVCTAD,20$      ;GET CURR INT VECT ADR
1250 006214' 016346 000004          MOV   PFWADR(R3),-(SP) ;STORE FLGWD ADR TO IDENTIFY ME
1251 006220' 004577 171646          JSR   RS,ATSTVEC      ;DO I HAVE VECTOR CONTROL?
1252 006224' 000000 20$: .WORD XXXX          ;MPG WILL TELL ME SINCE I CAN'T
1253 006226' 175756          .WORD RDINT-.        ;GET AT LOWER MEM IF MEM MGMT
1254 006230' 000401          BR    TRVEXT          ;BR IF I DONT HAVE CONTROL
1255 006232' 005725          TST   (RS)+          ;BYPASS BR INST IN S/R CALL
1256 006234' 000205  TRVEXT: RTS   R5          ;EXIT IN-LINE
1257
1258                                     ;TEST WRITE INTERRUPT VECTOR S/R
1259
1260 006236' 016767 171564 000016 TWVECT: MOV   IVCTAD,20$      ;GET CURR INT VECT ADR
1261 006244' 062767 000004 000010  ADD   #4,20$         ;ADJUST FOR WRITE INT

```

1262	006252'	016346	000004		MOV	PFWADR(R3),-(SP)	: STORE FLGWD ADR TO IDENTIFY ME
1263	006256'	004577	171610		JSR	RS,@TSTVEC	: DO I HAVE VECTOR CONTROL?
1264	006262'	000000		20S:	.WORD	XXXX	: MPG WILL TELL ME SINCE I CAN'T
1265	006264'	175402			.WORD	WRINT-	: GET AT LOWER MEM IF MEM MGMT
1266	006266'	000401			BR	TWEXT	: BR IF I DONT HAVE CONTROL
1267	006270'	005725			TST	(RS)+	: BYPASS BR INST IN S/R CALL
1268	006272'	000205		TWEXT:	RTS	RS	: RETURN IN-LINE
1269							
1270							

Handwritten scribble

```

1272                                     .SBTTL MESSAGE STORAGE AREA
1273
1274
1275                                     .NLIST BEX
1276
1277                                     .EVEN
1278 006274' 021520          PNMMSG: .ASCII /P#/
1279 006276' 054130          011      PROGM: .ASCII /XX/<011>
1280 006301' 101 020124 040514 ATMSG: .ASCII /AT LAST INT:/
1281 006315' 103 051125 042522 CURMSG: .ASCII /CURRENTLY:/
1282 006327' 105 042116 047440 RENDMG: .ASCII /END OF REPORT/
1283                                     .EVEN
1284 006344' 025052 025052 042040 UNITMG: .ASCII /*** DQ11 UNIT: /
1285 006364' 054130 054130 054130 UNASCI: .ASCII /XXXXXX/
1286 006372' 054130 054130 020075 DVRGMG: .ASCII /XXX= /
1287 006400' 054130 054130 054130 DVRGDT: .ASCII /XXXXXX/
1288
1289 006406' 054502 042524 035123 CNTSMG: .ASCII /BYTES: RD= /
1290 006422' 054130 054130 054130 BCMRD: .ASCII /XXXXXXXXXXXXX WR= /
1291 006444' 054130 054130 054130 BCMWR: .ASCII /XXXXXXXXXXXXX/
1292 006460' 005015          CRLF: .ASCII <015><012>
1293
1294 006462' 041411 047115 051504          .ASCII <011>/CMDS: RD= /
1295 006477' 130 054130 054130 CHDCRD: .ASCII /XXXXXX WR= /
1296 006512' 054130 054130 054130 CHDCWR: .ASCII /XXXXXX MISC= /
1297 006527' 130 054130 054130 CHDCMS: .ASCII /XXXXXX/<015><012>
1298 006537' 011 047111 042524          .ASCII <011>/INTERRUPTS: RD= /
1299 006561' 130 054130 054130 RDINMS: .ASCII /XXXXXX WR= /
1300 006574' 054130 054130 054130 WRINMS: .ASCII /XXXXXX/<015><012>
1301
1302 006604' 042411 051122 051117 ERRMSG: .ASCII <011>/ERRORS: VRC= /
1303 006623' 130 054130 054130 ERCVRC: .ASCII /XXXXXX BCC= /
1304 006640' 054130 054130 054130 ERCBCC: .ASCII /XXXXXX RNEM= /
1305 006655' 130 054130 054130 ERRNEM: .ASCII /XXXXXX TNEM= /
1306 006672' 054130 054130 054130 ERTNEM: .ASCII /XXXXXX/<015><012>
1307 006702' 004411 046122 052101          .ASCII <011><011>/RLAT= /
1308 006712' 054130 054130 054130 ERRLAT: .ASCII /XXXXXX TLAT= /
1309 006727' 130 054130 054130 ERTLAT: .ASCII /XXXXXX RCLK= /
1310 006744' 054130 054130 054130 ERRCLK: .ASCII /XXXXXX TCLK= /
1311 006761' 130 054130 054130 ERTCLK: .ASCII /XXXXXX/<015><012>
1312 006771' 011 040504 040524          .ASCII <011>/DATA ERRORS = /
1313 007010' 054130 054130 054130 ERCDTA: .ASCII /XXXXXX/
1314          007016'          CNTSEN=
1315                                     .EVEN
1316
1317                                     .LIST BEX
1318
1319          007016'          DVREND= .

```

F03

MAINDEC-11-DTDQA-B
DTDQAB.P11

DQ11 DEVICE ROUTINE FOR MPG
FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

MACY11 27(732) 24-SEP-76 14:09 PAGE 8

SEQ 0093

```

1321          .SBTTL FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES
1322
1323          ; PROGRAM TABLE FORMAT
1324
1325          000242 PTLGTH= 162. ;PROGRAM TABLE LENGTH - NON MEM MGMNT VERSION OF MPG
1326
1327          ;(PTLGTH= 212. ;PROGRAM TABLE LENGTH - MEM MGMNT VERSION OF MPG)
1328
1329          000000 PFLGWD= +0. ;PROGRAM FLAG WORD - 1 WORD
1330
1331          000002 URSTOP= 2 ; 1 = USER HAS STOPPED THIS PROGRAM
1332          000004 ERSTOP= 4 ; 1 = AN ERROR HAS STOPPED THIS PROGRAM
1333          000010 WT4IOT= 10 ; 1 = WAITING FOR I/O TERMINATION
1334          000020 CTPRIO= 20 ; 1 = CONSOLE OR PRINTER I/O IN PROGRESS
1335          000040 SETDED= 40 ; 1 = THIS PROG SET THE PRT DEV DEDICATED FLAG
1336          000100 OCPRES= 100 ; 1 = OBJ CODE IS PRESENT
1337          000200 USEUBM= 200 ; 1 = THIS PROG USES THE UNIBUS MAP (MEM MGMNT ONLY)
1338          100000 ACTIVE= 100000 ; 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)
1339
1340          000002 POPSW= +2. ;PROGRAM'S OPERATION SWITCHES - 1 WORD
1341
1342          100000 STONER= 100000 ; 1 = STOP PROG EXECUTION UPON ERROR
1343          040000 CYCPRG= 40000 ; 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
1344          020000 PRONER= 20000 ; 1 = DO NOT PRINT ON ERROR
1345          010000 BIT12= 10000 ; 0 = NOT USED
1346          004000 BIT11= 4000 ; 0 = NOT USED
1347          002000 CYCDV= 2000 ; 1 = CYCLE THE DEVICE LIST
1348          001000 GTNXTD= 1000 ; 1 = CYCLE ON SAME DEVICE UPON ERROR
1349          000400 DOERCK= 400 ; 1 = DON'T DO ERROR CHECKING
1350          000200 SPOPER= 200 ; 1 = DEVICE SPECIAL OPERATION
1351          000100 BIT6= 100 ; 0 = NOT USED
1352          000040 DOIOT= 40 ; 1 = DO NOT PERFORM I/O TIMEOUT
1353          000020 AUTORP= 20 ; 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
1354          000010 AURPEP= 10 ; 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
1355          000004 HSKPEP= 4 ; 1 = HOUSEKEEP COUNTS ONLY AT RUN COMMAND
1356          000002 PFBBOV= 2 ; 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
1357          000001 NOCOMP= 1 ; 1 = DO NOT PRINT PROG COMPLETED MSG
1358
1359          000004 PFWADR= +4. ;*;PROGRAM FLAGWORD ADDRESS - 1 WORD
1360
1361          000006 PASCIN= +6. ;PROGRAM'S NUMBER IN ASCII - 1 WORD
1362
1363          000010 PNAME= +8. ;PROGRAM'S NAME IN ASCII - 6 BYTES
1364
1365          000016 PRDIOA= +14. ;ADDRESS OF READ I/O AREA - 1 WORD
1366
1367          000020 PWRIOA= +16. ;ADDRESS OF WRITE I/O AREA - 1 WORD
1368
1369          000022 PSRCST= +18. ;SOURCE STATEMENTS START ADDRESS - 1 WORD
1370
1371          000024 POBJST= +20. ;OBJECT CODE START ADDRESS - 1 WORD
1372
1373          000026 PLNGTH= +22. ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD
1374
1375          000030 PTOCNT= +24. ;I/O TIMEOUT COUNT - 1 WORD
1376

```


1377	000032	PMDLCD= +26.	;DEV ROUT MODEL # CODE - 1 WORD
1378			
1379	000034	PDPNTR= +28.	;CURRENT DEVICE NUMBER POINTER - 1 BYTE
1380			
1381	000035	PCURDV= +29.	;CURRENT DEVICE # - 1 BYTE
1382			
1383	000036	PDNUMS= +30.	;DEVICE NUMBERS - 16 BYTES
1384			
1385	000056	PTEMO= +46.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1386			
1387	000060	PTEM1= +48.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1388			
1389	000062	PTEM2= +50.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1390			
1391	000064	PTEM3= +52.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1392			
1393	000066	PTEM4= +54.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1394			
1395	000070	PTEM5= +56.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1396			
1397	000072	PTEM6= +58.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1398			
1399	000074	PTEM7= +60.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1400			
1401	000076	PTEM8= +62.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1402			
1403	000100	PTEM9= +64.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1404			
1405	000102	PTEM10= +66.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1406			
1407	000104	PTEM11= +68.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1408			
1409	000106	PTEM12= +70.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1410			
1411	000110	PTEM13= +72.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1412			
1413	000112	PTEM14= +74.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1414			
1415	000114	PTEM15= +76.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1416			
1417	000116	PNBR= +78.	;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
1418			
1419	000120	PSRC= +80.	;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
1420			
1421	000122	PDST= +82.	;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
1422			
1423	000124	PSTKCT= +84.	;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
1424			
1425	000126	PSTKSV= +86.	;STACK WORDS STORAGE AREA - 30 WORDS
1426			
1427	000222	PSVREG= +146.	;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
1428			
1429	000236	PUSRPC= +158.	;USER'S CURRENT PROGRAM COUNTER - 1 WORD
1430			

```

1432                ;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMNT VERSION
1433
1434                ;(PRDIOX= +160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
1435
1436                ;(PRDIOV= +164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
1437
1438                ;(PWRIOX= +168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
1439
1440                ;(PWRIOV= +172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
1441
1442                ;(PUPARS= +176. ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)
1443
1444                ;(PUPDRS= +192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
1445
1446                ;(PUBMAP= +208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)
1447
1448                ;END OF MEM MGMNT ONLY ENTRIES
1449
1450                000240        PFSIZE= +160. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMNT
1451
1452                ;(PFSIZE= +210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMNT VERSION)
1453
1454                000242        PTEND= +162. ;END OF PROGRAM TABLE - NON MEM MGMNT VERSION
1455
1456                ;(PTEND= +212. ;END OF PROGRAM TABLE - MEM MGMNT VERSION)

```

```

1458           ;      DEVICE ROUTINE TABLE
1459
1460           ;
1461           000116      DRTLTH= 78.      ;DEVICE ROUTINE TABLE LENGTH
1462           ;
1463           ;
1464           000000      DEVRSZ= +0.      ;DEVICE ROUTINE SIZE IN BYTES - 1 WORD
1465           ;
1466           000002      DEVFWD= +2.      ;DEVICE ROUTINE FLAGWORD - 1 WORD
1467           ;
1468           000004      DEVIW1= +4.      ;DEVICE INTERFACE WORD # 1 - 1 WORD
1469           ;
1470           000006      DEVIW2= +6.      ;DEVICE INTERFACE WORD # 2 - 1 WORD
1471           ;
1472           000010      DEVIW3= +8.      ;DEVICE INTERFACE WORD # 3 - 1 WORD
1473           ;
1474           000012      DEVIW4= +10.     ;DEVICE INTERFACE WORD # 4 - 1 WORD
1475           ;
1476           000014      DEVIW5= +12.     ;DEVICE INTERFACE WORD # 5 - 1 WORD
1477           ;
1478           000016      DEVIW6= +14.     ;DEVICE INTERFACE WORD # 6 - 1 WORD
1479           ;
1480           000020      DEVIW7= +16.     ;DEVICE INTERFACE WORD # 7 - 1 WORD (SIZE)
1481           ;
1482           000022      DEVIW8= +18.     ;DEVICE INTERFACE WORD # 8 - 1 WORD (ERR)
1483           ;
1484           000024      DEVDRA= +20.     ;DEVICE REGISTERS ADDRESS - 1 WORD
1485           ;
1486           000026      DEVIVA= +22.     ;DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
1487           ;
1488           000030      DEVRPS= +24.     ;DEVICE READ PROCESSOR STATUS WORD (BUS REQ) - 1 WORD
1489           ;
1490           000032      DEVWPS= +26.     ;DEVICE WRITE PROC STATUS WORD (BUS REQ) - 1 WORD
1491           ;
1492           000034      DHKPAD= +28.     ;DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
1493           ;
1494           000036      DERPAD= +30.     ;DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
1495           ;
1496           000040      DKILAD= +32.     ;DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
1497           ;
1498           000042      DECTAD= +34.     ;DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
1499           ;
1500           000044      DTOEAD= +36.     ;DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
1501           ;
1502           000046      DEVI08= +38.     ;DEVICE I/O BUSY BRANCH ADDRESS (CIOBSY) - 1 WORD
1503           ;
1504           000050      DEVDER= +40.     ;DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
1505           ;
1506           000052      DVUPRT= +42.     ;USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
1507           ;
1508           000054      DVCPRN= +44.     ;CMND MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
1509           ;
1510           000056      DEVBTA= +46.     ;CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
1511           ;
1512           000060      DVBTDA= +48.     ;CONVERT BINARY TO DECIMAL ASCII BR ADR (BTASLZ) - 1 WORD
1513

```

1514	000062	DVPDTA= +50.	;CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
1515			
1516	000064	DVSFWD= +52.	;MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
1517			
1518	000066	DVSVEC= +54.	;SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
1519			
1520	000070	DVCVEC= +56.	;CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
1521			
1522	000072	DVTVEC= +58.	;TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
1523			
1524	000074	DVRINT= +60.	;RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
1525			
1526	000076	DVGETB= +62.	;GET DATA BYTE BR ADR (GETBYT) - 1 WORD
1527			
1528	000100	DVPUTB= +64.	;PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
1529			
1530	000102	DEVSTP= +66.	;DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
1531			
1532	000104	DEVETP= +68.	;DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
1533			
1534	000106	DVPTEP= +70.	;PACK TABLE EXTEN. REL POINTER - 1 WORD
1535			
1536	000110	DVVTEP= +72.	;VECTOR TABLE EXTEN. REL POINTER - 1 WORD
1537			
1538	000112	DVCTEP= +74.	;COMPILER TBL EXTEN. REL POINTER - 1 WORD
1539			
1540	000114	DVIWSP= +76.	;DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
1541			
1542	000116	DRTEND= +78.	;END OF DEVICE ROUTINE TABLE
1543			
1544			
1545			
1546	000001	.END	

ABBREV	005006R	002	DEVIW5=	000014	002	EMSGHD	005332R	002	LLISEN	001122R	002	PTEM3 =	000064	
ACTIVE=	100000		DEVIW6=	000016		ERCBC	006640R	002	LNIDLE	001122R	002	PTEM4 =	000066	
ANSWER	003330R	002	DEVIW7=	000020		ERCDA	007010R	002	LNWAIT	001122R	002	PTEM5 =	000070	
ATMSG	006301R	002	DEVIW8=	000022		ERCDB	005246R	002	LOCZ	000000R	002	PTEM6 =	000072	
AURPEP=	000010		DEVRS=	000030		ERCVC	006623R	002	LRECV	001122R	002	PTEM7 =	000074	
AUTORP=	000020		DEVRSZ=	000000		ERDIRG	005156R	002	LEND	001122R	002	PTEM8 =	000076	
BCCNT	001232R	002	DEVSTP=	000102		ERMBCT	005154R	002	LSLREG	001123R	002	PTEM9 =	000100	
BCMRD	006422R	002	DEVWPS=	000032		ERR	000022R	002	LSLSEQ	001123R	002	PTEND =	000242	
BCMWR	006444R	002	DHKPAD=	000034		ERRCLK	006744R	002	LSTATS	001122R	002	PTLGTH=	000242	
BINASC	000056R	002	DIDIST	001414R	002	ERRDIS	005064R	002	LWAIT	001122R	002	PTCNT =	000030	
BIT11 =	004000		DISCNT	004650R	002	ERREXT	003120R	002	MISC	001134R	002	PTSIZE=	000240	
BIT12 =	010000		DISCON	003422R	002	ERRFLG	003664R	002	MISCNT	001222R	002	PUSRPC=	000236	
BIT6 =	000100		DISPIS	005704R	002	ERRINT	004114R	002	MSADDR	003652R	002	PUTBYT	000100R	002
BTASLZ	000060R	002	DISPST	005712R	002	ERRLAT	006712R	002	NOCOMP=	000001		PMRIOA=	000020	
BYRD	001206R	002	DISUIV	006040R	002	ERRMSG	006604R	002	NOIDLE	003632R	002	RABORT	002212R	002
BYWR	001212R	002	DISUML	006066R	002	ERRNEM	006655R	002	NOWAIT	003134R	002	RANG	003410R	002
CALL	003330R	002	DISUNM	006002R	002	ERRRPT	005054R	002	OCPRES=	000100		RBUSRQ	000030R	002
CALLOK	003360R	002	DISUPR	006060R	002	ERRSNM	005164R	002	PASCIN=	000006		RCLKCT	001244R	002
CIOSBY	000046R	002	DKILAD=	000040		ERRTST	003062R	002	PC	=%000007		RCSR	001126R	002
CLIST	000054R	002	DOERCK=	000400		ERSTOP=	000004		PCURDV=	000035		RDCNT	001216R	002
CLRVCT=	000002		DOIOT =	000040		ERTBEN	005146R	002	PONUMS=	000036		RDNMT	002216R	002
CLRVEC	000070R	002	DREGAD	000024R	002	ERTCLK	006761R	002	POPNTN=	000034		RDINMS	006561R	002
CLWVCT=	000004		DRTEND=	000116		ERTLAT	006727R	002	PDST =	000122		RDINT	004204R	002
CMDCMS	006527R	002	DRTLTH=	000116		ERTNEM	006672R	002	PFBBOV=	000002		RDINCL	001700R	002
CMDCRD	006477R	002	DRWAIT=	100000		FDUPLX	003172R	002	PFLGWD=	000000		RDINST	001706R	002
CMDCHR	006512R	002	DTOEAD=	000044		FLAG	001254R	002	PFWADR=	000004		RDNMT	001754R	002
CNTSEN=	007016R	002	DVBTDA=	000060		FLAGWD	000002R	002	PLNGTH=	000026		RDPBC	003660R	002
CNTSMG	006406R	002	DVCMS	000306R	002	GETBYT	000076R	002	PMDLCD=	000032		RDPBSY=	010000	
CRESET	003210R	002	DVCPT=	000054		GTNXTD=	001000		PNAME =	000010		RDRPRM	002012R	002
CRLF	006460R	002	DVCPT=	000740R	002	GVECAD	004076R	002	PNAME =	000116		RDSBC	003662R	002
CSTAT	001136R	002	DVCTEP=	000112		HANGUP	003416R	002	PNMMSG	006274R	002	RDSBSY=	004000	
CSYSFW	000064R	002	DVCVEC=	000070		HUPLX	003154R	002	POBJST=	000024		RDESEC	002074R	002
CTPRIO=	000020		DVGETB=	000076		HNGDL1	003434R	002	POPSW =	000002		RDTERM	004356R	002
CUPGER	000050R	002	DVIWSP=	000114		HNGDL2	003440R	002	PRDIOA=	000016		READ	001474R	002
CURMSG	006315R	002	DVIWST	001120R	002	HSKEEP	004466R	002	PRERLN	004766R	002	RCV	003522R	002
CWRINT	001440R	002	DVMVTE	000634R	002	HSKPEN=	001256R	002	PRINT	006072R	002	RELEAS	003124R	002
CYCDVL=	002000		DVPDTA=	000062		HSKPEP=	000004		PROGNM	006276R	002	RENDMG	006327R	002
CYCPRG=	040000		DVPKTE	000424R	002	HSKPST=	001126R	002	PRONER=	020000		REPORT	004522R	002
DATAER	001250R	002	DVPTEP=	000106		IDLE	003612R	002	PRTEX	006204R	002	REPTBL	005010R	002
DECASC	000062R	002	DVPUTB=	000100		INTEXT	004104R	002	PSRC =	000120		RERR	001132R	002
DECTAD=	000042		DVREGE	000146R	002	ISTAT =	001126R	002	PSRCST=	000022		RERVEC	004404R	002
DERPAD=	000036		DVREGS	000116R	002	IVCTAD	000026R	002	PSTKCT=	000124		RESREG	005450R	002
DEVBTA=	000056		DVREND=	007016R	002	KILL	001370R	002	PSTKSV=	000126		RETN	003132R	002
DEVDER=	000050		DVREX	004756R	002	KILLEX	001466R	002	PSVREG=	000222		RICNT	001224R	002
DEVDRR=	000024		DVRGDT	006400R	002	LANSWR	001122R	002	PTEM0 =	000056		RLATCT	001240R	002
DEVETP=	000104		DVRGMG	006372R	002	LCALL	001122R	002	PTEM1 =	000060		RNEMCT	001234R	002
DEVFWD=	000002		DVRINT=	000074		LCOUNT	001122R	002	PTEM10=	000102		RNOABT=	002000	
DEVI0B=	000046		DVSFWD=	000064		LCRST	001122R	002	PTEM11=	000104		RPDONE	004304R	002
DEVIVA=	000026		DVSVEC=	000066		LFULL	001122R	002	PTEM12=	000106		RPSBSY	001774R	002
DEVIW1=	000004		DVTVEC=	000072		LHALF	001122R	002	PTEM13=	000110		RPTBAS	004714R	002
DEVIW2=	000006		DVUPRT=	000052		LHNGUP	001122R	002	PTEM14=	000112		RPTEND	004746R	002
DEVIW3=	000010		DVVTEP=	000110		LIDLE	001122R	002	PTEM15=	000114		RPTLP	004676R	002
DEVIW4=	000012		EMSGBF	005346R	002	LISEN	003366R	002	PTEM2 =	000062		RRINTV	004424R	002

RSETUP	002156R	002	SETVEC	000066R	002	TLATCT	001242R	002	UNASCI	006364R	002	WRITE	002234R	002
RTNINT	000074R	002	SISTAT	005506R	002	TLOOP	002346R	002	UNITMG	006344R	002	WRMCL	002414R	002
RWINTV	004442R	002	SIZE	000020R	002	TMAINT	001632R	002	URSTOP	= 000002		WRMNST	002422R	002
R0	=%000000		SP	=%000006		TNEMCT	001236R	002	USEUBM	= 000200		WRNOWT	002470R	002
R1	=%000001		SOPER	= 000200		TOECNT	001252R	002	VRCNT	001230R	002	WRPBC	003654R	002
R2	=%000002		SRDVEC	001544R	002	TOEMSG	001344R	002	WABORT	002722R	002	WRPBSY	= 040000	
R3	=%000003		STMNMG	005416R	002	TOUTER	001256R	002	WAIT	002742R	002	WRPRIM	002526R	002
R4	=%000004		STMNT	005330R	002	TOUTEX	001336R	002	WBUSRQ	000032R	002	WRSBC	003656R	002
R5	=%000005		STNUM	005426R	002	TRMIST	003002R	002	WICNT	001226R	002	WRSBSY	= 020000	
SAVREG	005434R	002	STONER	= 100000		TRVECT	006206R	002	WNOABT	= 001000		WRSEC	002612R	002
SECREG	005702R	002	STSTAT	005562R	002	TRVEXT	006234R	002	WPDONE	004000R	002	WRTERM	004056R	002
SELREG	003542R	002	SUPTAD	005466R	002	TSTIEB	002764R	002	WPSBSY	002510R	002	WSETUP	002674R	002
SELSEQ	003556R	002	SWRVEC	002306R	002	TSTVEC	000072R	002	WRCNT	001220R	002	WT4IOT	= 000010	
SEND	003464R	002	TAILOR	005716R	002	TWVECT	006236R	002	WRDNWT	002726R	002	WVCLR	001572R	002
SENDOK	003514R	002	TCLKCT	001246R	002	TWVEXT	006272R	002	WRINMS	006574R	002	XXXX	= 000000	
SETDED	= 000040		TCSR	001130R	002	ULIST	000052R	002	WRINT	003666R	002	.	= 007016R	002

. ABS. 000000 000
000000 001
DQ11 007016 002

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*, DTDQAB/NL: TOC/DOC=DTDQAB.P11
RUN-TIME: 4 9 1 SECONDS
RUN-TIME RATIO: 21/15=1.3
CORE USED: 5K (9 PAGES)

DOCUMENT PAGES: 37

M03

Spooler runtime 5 Seconds, 26 KCS, 147 disk reads, 4 disk writes, 37 pages

00111111111111111111111111111111110
000000011111111111222222222223333333333344444444444555555555556666666666677777777777888888888889999999999900000000000111111111112222222222233312
000111111111111111111111111111111110
0000000111111111122222222222333333333334444444444455555555555666666666667777777777788888888888999999999990000000000111111111111111111111112222222222233312

567890123456789012 **