

TM11

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

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

NOV 1976
digital
MADE IN U.S.A.

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
110
111
112

.SBTTL STANDARD DEVICE ROUTINE TABLE

.TITLE MAINDEC-11-DTTMA-B TM11/TUID DEVICE ROUTINE FOR MPG

;REVISION 'B'

;FILENAME OF "TTMABO.MPG" ON MPG/XXDP MEDIA

;MACY11: DTTMA?,DTTMA?/CRF:SYM/DOC=DTTMA?.P11

;LNKX11: DTTMA?.MPG/B:0+DTTMA?/E

;PAPER TAPE: PUNCH DTTMA?.MPG/FILE:ELEV

000000'

.CSECT TM11

.DSABL GBL

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

000000' 005340
000002' 000000

LOCZ: .WORD DVREND-
DFLGWD: .WORD 0

:DEVICE ROUT SIZE IN BYTES
:DEVICE ROUT FLAGWORD
: BIT 15 = "NOWAIT" FLAG
: BIT 14 = BPI DEN 8 BIT
: BIT 13 = BPI DEN 5 BIT
: BIT 11 = EVEN PARITY BIT
: BIT 3 = ROLLBACK EXH. FLAG
: BIT 1 = DO I/O TERMINATION
: BIT 0 = ERROR ON I/O CMND
: # OF ROLLBACKS FOR READ
: # OF ROLLBACKS FOR WRITE
: 1 = EOF ENCOUNTERED
: 1 = EOT ENCOUNTERED
: 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)
: NOT USED
: 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
: CONVERT PACKED DECIMAL TO ASCII BR ADR
: MPG SYSTEM FLAGWORD ADR
: SET INT VECT ROUT BR ADR

000004' 000000
000006' 000000
000010' 000000
000012' 000000
000014' 000000
000016' 000000
000020' 000001
000022' 000000
000024' 172520
000026' 000224
000030' 000240
000032' 000000
000034' 001022
000036' 001076
000040' 001524
000042' 000762
000044' 001420
000046' 000000
000050' 000000
000052' 000000
000054' 000000
000056' 000000
000060' 000000
000062' 000000
000064' 000000
000066' 000000

RDRB: .WORD 0
WRRB: .WORD 0
EOF: .WORD 0
EOT: .WORD 0
: .WORD 0
: .WORD 0
SIZE: .WORD 1
ERR: .WORD 0
DREGAD: .WORD 172520
IVCTAD: .WORD 224
PSWD: .WORD 240
: .WORD 0
: .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
DECASC: .WORD 0
CSYSFW: .WORD 0
SETVEC: .WORD 0

113	000070'	000000		CLAVEC:	.WORD	0		: CLEAR INT VECTOR ROUT BR ADR
114	000072'	000000		TSTVEC:	.WORD	0		: TEST INT VECTOR ROUT BR ADR
115	000074'	000000		RTNINT:	.WORD	0		: RETURN FROM INT ROUT BR ADR
116	000076'	000000		GETBYT:	.WORD	0		: GET DATA BYTE ROUT BR ADR
117	000100'	000000		PUTBYT:	.WORD	0		: PUT DATA BYTE ROUT BR ADR
118	000102'	000014			.WORD	DVREGS-		: ADR OF DEVICE REGISTER NAMES
119	000104'	000056			.WORD	DVCMD5-		: ADR OF DEVICE FUNCTIONS
120	000106'	000156			.WORD	DVPKTE-		: ADR OF PACK TBL EXTENSION
121	000110'	000334			.WORD	DVMVTE-		: ADR OF MODEL VECTOR TBL EXTEN.
122	000112'	000422			.WORD	DVCPT5-		: ADR OF COMPILER TBL EXTEN.
123	000114'	000564			.WORD	DVIWST-		: ADR OF DEV INTERFACE WD SYM TBL
124								
125								
126	000116'	052115	020123	DVREGS:	.ASCII	/MTS /		: VALID DEVICE REGISTER NAMES &
127	000122'	000000			.WORD	0		: THEIR POSITIONS RELATIVE TO
128	000124'	052115	020103		.ASCII	/MTC /		: THE DEVICE REGISTERS BASE ADDRESS.
129	000130'	000002			.WORD	2		
130	000132'	041115	041522		.ASCII	/MBRC/		
131	000136'	000004			.WORD	4		
132	000140'	041515	040515		.ASCII	/MCMA/		
133	000144'	000006			.WORD	6		
134	000146'	052115	020104		.ASCII	/MTD /		
135	000152'	000010			.WORD	10		
136	000154'	052115	042122		.ASCII	/MTRD/		
137	000160'	000012			.WORD	12		
138		000162'		DVREGE=	.			
139								
140	000162'	120	201	DVCMD5:	.BYTE	120,201		: VALID DEVICE FUNCTIONS
141	000164'	001600			.WORD	READ-		: FLAG BYTE:
142	000166'	130	201		.BYTE	130,201		: BIT 7 = NPR DEV
143	000170'	001624			.WORD	WRITE-		: BIT 3 = MASSBUS DEV
144	000172'	376	000		.BYTE	376,0		: BIT 0 = 2 WORDS FOR ADR
145	000174'	001436			.WORD	NOWAIT-		: (18 BIT ADRS)
146	000176'	375	000		.BYTE	375,0		
147	000200'	001412			.WORD	WAIT-		
148	000202'	374	000		.BYTE	374,0		
149	000204'	000730			.WORD	REPORT-		
150	000206'	373	000		.BYTE	373,0		
151	000210'	000724			.WORD	REPORT-		
152	000212'	372	000		.BYTE	372,0		
153	000214'	001730			.WORD	CRESET-		
154	000216'	371	201		.BYTE	371,201		
155	000220'	001624			.WORD	WRETRG-		
156	000222'	370	000		.BYTE	370,0		
157	000224'	001632			.WORD	WREOF-		
158	000226'	367	000		.BYTE	367,0		
159	000230'	001640			.WORD	SPFWD-		
160	000232'	366	000		.BYTE	366,0		
161	000234'	001656			.WORD	SPREV-		
162	000236'	365	000		.BYTE	365,0		
163	000240'	001660			.WORD	REWIND-		
164	000242'	364	000		.BYTE	364,0		
165	000244'	001666			.WORD	OFFLIN-		
166	000246'	363	000		.BYTE	363,0		
167	000250'	001402			.WORD	EVEN-		
168	000252'	362	000		.BYTE	362,0		

169	000254'	001366			.WORD	ODD-	
170	000256'	361	000		.BYTE	361,0	
171	000260'	001402			.WORD	BPI-	
172	000262'	177777			.WORD	177777	;TABLE TERMINATOR
173							
174	000264'	047516	040527	052111	DVPKTE: .ASCII	/NOWAIT/	;PACK TABLE EXTENSION
175	000272'	376	000		.BYTE	376,0	
176	000274'	020040	040527	052111	.ASCII	/ WAIT/	
177	000302'	375	000		.BYTE	375,0	
178	000304'	052123	052101	051525	.ASCII	/STATUS/	
179	000312'	374	000		.BYTE	374,0	
180	000314'	047503	047125	051524	.ASCII	/COUNTS/	
181	000322'	373	000		.BYTE	373,0	
182	000324'	051103	051505	052105	.ASCII	/CRESET/	
183	000332'	372	000		.BYTE	372,0	
184	000334'	051127	044505	043522	.ASCII	/WREIRG/	
185	000342'	371	000		.BYTE	371,0	
186	000344'	053440	042522	043117	.ASCII	/ WREOF/	
187	000352'	370	000		.BYTE	370,0	
188	000354'	051440	043120	042127	.ASCII	/ SPFWD/	
189	000362'	367	000		.BYTE	367,0	
190	000364'	051440	051120	053105	.ASCII	/ SPREV/	
191	000372'	366	000		.BYTE	366,0	
192	000374'	042522	044527	042116	.ASCII	/REWIND/	
193	000402'	365	000		.BYTE	365,0	
194	000404'	043117	046106	047111	.ASCII	/OFFLIN/	
195	000412'	364	000		.BYTE	364,0	
196	000414'	020040	053105	047105	.ASCII	/ EVEN/	
197	000422'	363	000		.BYTE	363,0	
198	000424'	020040	047440	042104	.ASCII	/ ODD/	
199	000432'	362	000		.BYTE	362,0	
200	000434'	020040	041040	044520	.ASCII	/ BPI/	
201	000442'	361	000		.BYTE	361,0	
202							
203	000444'	000376	000732		DVMVTE: .WORD	376, LNWAIT-LOCZ	;MODEL VECTOR TABLE EXTEN.
204	000450'	000375	000732		.WORD	375, LWAIT-LOCZ	
205	000454'	000374	000732		.WORD	374, LSTATS-LOCZ	
206	000460'	000373	000732		.WORD	373, LCOUNT-LOCZ	
207	000464'	000372	000732		.WORD	372, LCRST-LOCZ	
208	000470'	000371	000733		.WORD	371, LWEIRG-LOCZ	
209	000474'	000370	000732		.WORD	370, LWREOF-LOCZ	
210	000500'	000367	000742		.WORD	367, LSPFWD-LOCZ	
211	000504'	000366	000742		.WORD	366, LSPREV-LOCZ	
212	000510'	000365	000732		.WORD	365, LRWIND-LOCZ	
213	000514'	000364	000732		.WORD	364, LOFFLN-LOCZ	
214	000520'	000363	000732		.WORD	363, LEVEN-LOCZ	
215	000524'	000362	000732		.WORD	362, LODD-LOCZ	
216	000530'	000361	000742		.WORD	361, LBPI-LOCZ	
217							
218							
219							
220							
221	000534'	003	376		DVCPTTE: .BYTE	3, 376	;NO WAIT
222	000536'	004537	000012		.WORD	4537, 10.	
223	000542'	003	375		.BYTE	3, 375	;WAIT
224	000544'	004537	000012		.WORD	4537, 10.	

225	000550'	004	374		.BYTE	4,374		;STATUS
226	000552'	004537	000012	001002	.WORD	4537,10.,1002		
227	000560'	004	373		.BYTE	4,373		;COUNTS
228	000562'	004537	000012	001001	.WORD	4537,10.,1001		
229	000570'	003	372		.BYTE	3,372		;CONTROL RESET
230	000572'	004537	000012		.WORD	4537,10.		
231	000576'	006	371		.BYTE	6,371		;WRITE EXTENDED INTER-RECORD GAP
232	000600'	004537	000012	000000	.WORD	4537,10.,0,2,2		
	000606'	000002	000002					
233	000612'	003	370		.BYTE	3,370		;WRITE END OF FILE
234	000614'	004537	000012		.WORD	4537,10.		
235	000620'	004	367		.BYTE	4,367		;SPACE FORWARD
236	000622'	004537	000012	000000	.WORD	4537,10.,0		
237	000630'	004	366		.BYTE	4,366		;SPACE REVERSE
238	000632'	004537	000012	000000	.WORD	4537,10.,0		
239	000640'	003	365		.BYTE	3,365		;REWIND
240	000642'	004537	000012		.WORD	4537,10.		
241	000646'	003	364		.BYTE	3,364		;OFFLINE
242	000650'	004537	000012		.WORD	4537,10.		
243	000654'	003	363		.BYTE	3,363		;EVEN
244	000656'	004537	000012		.WORD	4537,10.		
245	000662'	003	362		.BYTE	3,362		;ODD
246	000664'	004537	000012		.WORD	4537,10.		
247	000670'	004	361		.BYTE	4,361		;BPI
248	000672'	004537	000012	000000	.WORD	4537,10.,0		

...
DEVICE INTERFACE WORD SYMBOL TABLE

253	000700'	042122	041122	DVIWST:	.ASCII	/RDRB/		
254	000704'	000004			.WORD	DEVIW1		
255	000706'	051127	041122		.ASCII	/WRRB/		
256	000712'	000006			.WORD	DEVIW2		
257	000714'	047505	020106		.ASCII	/EOF /		
258	000720'	000010			.WORD	DEVIW3		
259	000722'	047505	020124		.ASCII	/EOT /		
260	000726'	000012			.WORD	DEVIW4		
261	000730'	177777			.WORD	177777		;END OF TABLE

...
MODEL STATEMENT TABLE EXTENSION

266	000732'			LNWAIT:				
267	000732'			LWAIT:				
268	000732'			LSTATS:				
269	000732'			LCOUNT:				
270	000732'			LWREOF:				
271	000732'			LRWIND:				
272	000732'			LOFFLN:				
273	000732'			LEVEN:				
274	000732'			LODD:				
275	000732'	000		LCRST:	.BYTE	0		
276	000733'	377	051106 046517	LWEIRG:	.ASCIZ	<377>/FROM/<377>		
	000740'	000377						
277	000742'			LSPFWD:				
278	000742'			LSPREV:				

279	000742'	377	000	LBPI:	.BYTE	377,0	
280					.EVEN		
281							
282		000744'		HSKPST=	.		
283	000744'	000000		ISTAT:	.WORD	0	;STORAGE FOR DEV REG'S AT INT
284	000746'	000000			.WORD	0	
285	000750'	000000			.WORD	0	
286	000752'	000000			.WORD	0	
287	000754'	000000			.WORD	0	
288	000756'	000000			.WORD	0	
289							
290	000760'	000006		CSTAT:	.BLKW	6	;DEV REG CURRENT VALUES STORAGE
291							
292	000774'	000000		BYRD:	.WORD	0	;BYTES READ COUNT (READ)
293	000776'	000000			.WORD	0	
294	001000'	000000		BYWR:	.WORD	0	;BYTES WRITTEN COUNT (WRITE & WREIRG)
295	001002'	000000			.WORD	0	
296	001004'	000000		RDCNT:	.WORD	0	;READ CMND COUNT (READ)
297	001006'	000000		WRCNT:	.WORD	0	;WRITE CMND COUNT (WRITE & WREIRG)
298	001010'	000000		MISCNT:	.WORD	0	;MISC. CMND COUNT (WREOF, SPFWD, SPREV, ;REWIND, OFFLIN, & CRESÉT)
299							
300	001012'	000000		RRBCNT:	.WORD	0	;# OF READ ROLLBACKS
301	001014'	000000		WRBCNT:	.WORD	0	;# OF WRITE ROLLBACKS
302	001016'	000000		EOFcnt:	.WORD	0	;# OF EOF'S
303	001020'	000000		EOTcnt:	.WORD	0	;# OF EOT'S
304	001022'	000000		ERRCNT:	.WORD	0	;DEVICE ERRORS COUNT
305	001024'	000000		DATAER:	.WORD	0	;DATA ERRORS COUNT
306	001026'	000000		INTCNT:	.WORD	0	;INTERRUPTS COUNT
307							
308	001030'	000000		TOECNT:	.WORD	0	;# OF ENTRIES INTO T/O ERROR ROUT
309	001032'	000000		TOEMAX:	.WORD	0	;MAX # OF TIMEOUTS
310	001034'	000000		ERRADR:	.WORD	0	;CURR ADR IN USER PROG
311	001036'	000000		CNTADR:	.WORD	0	;ADR OF BYTE COUNT TOTALS
312	001040'	000000		CURFLG:	.WORD	0	;FLAG WORD OF CURR CMND
313	001042'	000000		CURCNT:	.WORD	0	;BYTE CNT FOR CURR CMND
314	001044'	000000		FINCNT:	.WORD	0	;FINAL WORD COUNT (MBRC)
315	001046'	000000		RBCMD:	.WORD	0	;CURR CMND FOR ROLLBACK
316	001050'	000000		RBADR:	.WORD	0	;CURR ADR FOR ROLLBACK
317	001052'	000000		RBCNT:	.WORD	0	;CURR BYTE CNT FOR ROLLBACK
318	001054'	000000		NUMRB:	.WORD	0	;NUMBER OF ROLLBACKS ON CURR CMND
319		001056'		HSKPEN=	.		
320							
321		000000		XXXX=	0		;VALUE TO BE TAILORED BY DEV ROUT

323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378

.SBTTL TM11 SUPPORT ROUTINES ENTERED FROM MPG

;DEVICE ROUTINE HOUSEKEEPING

```

;JSR   R5,HSKEEP           S/R CALL
;.WORD 0 OR 1              0 = DO HSKP PER OPSW
;R2 = PROG'S OPSW         1 = UNCOND. DO HSKP
;DESTROYS R0,R1
    
```

```

HSKEEP: MOV   #60000,DFLGWD ;SET BPI TO 800 & PARITY TO ODD
        MOV   #1,RDRB      ;SET READ ROLLBACK CNT TO 1
        MOV   #3,WRRB      ;SET WRITE ROLLBACK CNT TO 3
        TST   (R5)+        ;UNCONDITIONALLY DO HSKP?
        BNE   10$          ;N,Y-10$
        BIT   #HSKPEP,R2   ;OPSW SPECIFY EACH PASS HSKP?
        BNE   30$          ;Y,N-30$
10$:    MOV   PC,R0         ;SET UP FIRST WD ADR
        ADD   #HSKPST-.,R0
        MOV   #HSKPEN-HSKPST/2,R1 ;SET UP # OF WORDS
20$:    CLR   (R0)+        ;HSKP ALL NECESSARY AREAS
        DEC   R1
        BNE   20$
30$:    RTS   R5           ;EXIT IN-LINE
    
```

;TM11 REPORT ROUTINE

```

;JSR   R5,REPORT          S/R CALL
;.WORD FLGWD              FLAGWORD
;BIT 15 = CMND MODE CALL
;BIT 9 = PROG STMT CALL
;BIT 1 = DO STATUS REPORT
;BIT 0 = DO COUNTS REPORT
    
```

```

REPORT: JSR   R0,SAVREG    ;SAVE REG'S R0 - R5
        BIT   #177776,(R5) ;DISPLAYING CNTS AT END OF
        BNE   10$         ;PROG PASS? (Y,N-10$)
        MOV   PC,R0       ;SET UP ADR OF CNTS
        ADD   #BYRD-.,R0
        MOV   #14.,R1
5$:    TST   (R0)+        ;GET # OF CNT WORDS
        BNE   10$         ;THIS CNT WORD = 0?
        DEC   R1          ;Y,N-10$
        BNE   5$         ;DECR WORD CNT
        BR   DVREX        ;CK'ED ALL WORDS? (Y,N-5$)
10$:   JSR   PC,SUPTAD    ;GO TO EXIT -- ALL CNTS ARE 0'S
        MOV   (R5)+,R4    ;SET UP PROG TBL ADR IN R3
        BIT   #2,R4       ;GET FLAGWORD
        BEQ   DISCNT      ;GOING TO DO STATUS DISPLAY?
        JSR   R5,STSTAT   ;Y,N-DISCNT
        .WORD CSTAT-      ;GO STORE STATUS REG'S
        MOV   PC,R0       ;SET UP ADR OF REG'S AT
        ADD   #I$TAT-.,R0 ;LAST INT
        MOV   #6,R1       ;SET UP # OF REG'S
    
```


379	001226'	005720		20\$:	TST	(R0)+		;ALL REG'S = 0?
380	001230'	001003			BNE	30\$;N,Y-40\$
381	001232'	005301			DEC	R1		
382	001234'	001374			BNE	20\$		
383	001236'	000412			BR	40\$		
384	001240'	004767	002702	30\$:	JSR	PC,DISUNM		;DISPLAY CURR UNIT #
385	001244'	004567	003114		JSR	R5,PRINT		;ISSUE 'AT LAST INT' MSG
386	001250'	003235			.WORD	ATIMSG-		
387	001252'	000014			.WORD	12.		
388	001254'	004567	002756		JSR	R5,DISPST		;GO DISPLAY STATUS AT LAST INT
389	001260'	177464			.WORD	ISTAT-		
390	001262'	000402			BR	45\$;CONTINUE DISPLAY
391	001264'	004767	002656	40\$:	JSR	PC,DISUNM		;DISPLAY CURR UNIT #
392	001270'	004567	003070	45\$:	JSR	R5,PRINT		;ISSUE 'CURRENTLY' MSG
393	001274'	003225			.WORD	CURMSG-		
394	001276'	000012			.WORD	10.		
395	001300'	004567	002732		JSR	R5,DISPST		;GO DISPLAY CURRENT STATUS
396	001304'	177454			.WORD	CSTAT-		
397	001306'	004767	003014		JSR	PC,PRTIWD		;GO DISPLAY INFO WORDS
398	001312'	000402			BR	DISCT1		;CHECK FOR COUNTS DISPLAY
399	001314'	004767	002626	DISCNT:	JSR	PC,DISUNM		;DISPLAY CURR UNIT #
400	001320'	032704	000001	DISCT1:	BIT	#1,R4		;DISPLAY COUNTS?
401	001324'	001431			BEQ	RPTEND		;Y,N-RPTEND
402	001326'	012700	000016		MOV	#14,R0		;SET UP # OF WORDS
403	001332'	010701			MOV	PC,R1		;SET UP ADR OF CNTS
404	001334'	062701	177440		ADD	#BYRD-. ,R1		
405	001340'	010702			MOV	PC,R2		;SET UP TBL ADR
406	001342'	062702	000066		ADD	#REPTBL-. ,R2		
407	001346'	012267	000012	RPTLP:	MOV	(R2)+,RPTBAS		;MOV MSG ADR TO S/R LINKAGE
408	001352'	004067	002464		JSR	R0,SAVREG		;SAVE ALL REG'S
409	001356'	011100			MOV	(R1),R0		;GET CURRENT COUNT
410	001360'	004577	176472		JSR	R5,JBINASC		;CONVERT IT TO ASCII
411	001364'	000000		RPTBAS:	.WORD	XXXX		
412	001366'	004067	002464		JSR	R0,RESREG		;RESTORE REG'S
413	001372'	005721			TST	(R1)+		;POINT AT NXT CNT
414	001374'	005300			DEC	R0		;DONE ALL WORDS?
415	001376'	001363			BNE	RPTLP		;Y,N-RPTLP
416	001400'	004567	002760		JSR	R5,PRINT		;GO ISSUE COUNTS MSG
417	001404'	003212			.WORD	CNTSMG-		
418	001406'	000330			.WORD	CNTSEN-CNTSMG		
419	001410'	004567	002750	RPTEND:	JSR	R5,PRINT		;ISSUE "END OF REPORT" MSG
420	001414'	003117			.WORD	RENDMG-		
421	001416'	177763			.WORD	-13.		
422	001420'	004067	002432	DVREX:	JSR	R0,RESREG		;RESTORE REGISTERS
423	001424'	005725			TST	(R5)+		;SET UP RETURN POINT
424	001426'	000205			RTS	R5		;EXIT IN-LINE
425								
426								
427	001430'	003246		REPTBL:	.WORD	BCMRD-RPTBAS		
428	001432'	003254			.WORD	BCMRD+6-RPTBAS		
429	001434'	003270			.WORD	BCMWR-RPTBAS		
430	001436'	003276			.WORD	BCMWR+6-RPTBAS		
431	001440'	003323			.WORD	CMDCRD-RPTBAS		
432	001442'	003336			.WORD	CMDCWR-RPTBAS		
433	001444'	003353			.WORD	CMDCMS-RPTBAS		
434	001446'	003404			.WORD	CNTRRB-RPTBAS		


```

435 001450' 003417          .WORD  CNTWRB-RPTBAS
436 001452' 003444          .WORD  CNTEOF-RPTBAS
437 001454' 003462          .WORD  CNTEOT-RPTBAS
438 001456' 003511          .WORD  CNTEOT-RPTBAS
439 001460' 003526          .WORD  CNTERR-RPTBAS
440 001462' 003554          .WORD  CNTDER-RPTBAS
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479

```

;TIMEOUT ERROR ROUTINE
;JSR R5,TOUTER S/R CALL
TOUTER: INC TOECNT ;INCR # OF TIMEOUTS THAT OCCURRED
CMPB TOECNT,TOEMAX+1 ;AT MAX # OF TIMEOUTS IN A ROW?
BNE TOUTEX ;Y,N-TOUTEX
JSR RO,SAVREG ;SAVE ALL REGISTERS
JSR PC,SUPTAD ;SET UP MTC & PROG TBL ADR'S
JSR R5,STSTAT ;STORE CURRENT STATUS
.WORD CSTAT-
JSR R5,TVECT ;DO I HAVE VECTOR CONTROL?
BR 10\$;BR IF I DON'T
MOVB #11,(R4) ;RESET INT ENABLE
JSR PC,RINTV ;RESET THE INTERRUPT VECTOR
10\$: BIC #WT4IOT,(R3) ;RESET WAITING FOR I/O FLAG
JSR R5,ERRCS1 ;ISSUE TIMEOUT ERROR MSG
.WORD IOT0-ERMBAS
.WORD 14.
JSR RO,RESREG ;RESTORE REGISTERS
MOV (SP)+,R5 ;REMOVE RETURN ADR
JMP @CUPGER ;GO TO ERROR EXIT
TOUTEX: RTS R5 ;EXIT IN-LINE

;KILL USER PROGRAM ROUTINE
;JSR R5,KILL S/R CALL
;R3 MUST CONTAIN PROG TBL ADR
;DESTROYS RO,R1
KILL: JSR R5,TVECT ;CK IF I HAVE VECTOR CONTROL
BR KILLEX ;BR IF I DON'T
MOV DREGAD,R1 ;GET DEV REG ADR
MOVB #11,2(R1) ;RESET INT ENABLE
JSR PC,RINTV ;RESET INT VECTOR INFO
KILLEX: RTS R5 ;EXIT IN-LINE

481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536

.SBTTL TM11 FUNCTION ROUTINES

;"WAIT" FUNCTION ROUTINE

;JSR R5, WAIT FUNCTION CALL

001612' 042767 100000 176162 WAIT: BIC #100000,DFLGWD ;RESET THE "NOWAIT" FLAG
001620' 004767 001252 JSR PC,CKDBSY ;WAIT IF BUSY & DO TERMINATION
001624' 004767 002136 JSR PC,RINTV ;RESET THE INTERRUPT VECTOR
001630' 000205 RTS R5 ;EXIT IN-LINE

;"NOWAIT" FUNCTION ROUTINE

;JSR R5,NOWAIT FUNCTION CALL

001632' 052767 100000 176142 NOWAIT: BIS #100000,DFLGWD ;SET THE "NOWAIT" FLAG
001640' 000205 RTS R5 ;EXIT IN-LINE

;"ODD" FUNCTION ROUTINE

;JSR R5,ODD FUNCTION CALL

001642' 042767 004000 176132 ODD: BIC #4000,DFLGWD ;RESET THE EVEN FLAG
001650' 000205 RTS R5 ;EXIT IN-LINE

;"EVEN" FUNCTION ROUTINE

;JSR R5,EVEN FUNCTION CALL

001652' 052767 004000 176122 EVEN: BIS #4000,DFLGWD ;SET THE EVEN FLAG
001660' 000205 RTS R5 ;EXIT IN-LINE

;"BPI" FUNCTION ROUTINE

;JSR R5,BPI FUNCTION CALL
;.WORD VALUE BPI BITS VALUE

001662' 004767 001274 BPI: JSR PC,STSADR ;STORE THIS STMT'S MEM ADR
001666' 012500 MOV (R5)+,R0 ;GET BIT VALUES
001670' 010701 MOV PC,R1 ;SET UP ADR OF VALID BPI VALUES
001672' 062701 000060 ADD #BPIVVL-.,R1
001676' 112102 10\$: MOV (R1)+,R2 ;GET VALID VALUE
001700' 100412 BMI BPIER ;END OF TBL? (N,Y-BPIER)
001702' 112103 MOV (R1)+,R3 ;GET CORRESP. BIT VALUES
001704' 020200 CMP R2,R0 ;MATCH THIS VALID VALUE?
001706' 001373 BNE 10\$;Y,N-10\$
001710' 042767 060000 176064 BIC #60000,DFLGWD ;RESET BPI BITS IN FLGWD
001716' 000303 SWAB R3 ;ALIGN BIT VALUES
001720' 050367 176056 BIS R3,DFLGWD ;SET IN NEW BPI BIT VALUES
001724' 000205 RTS R5 ;EXIT TO USER PROG
001726' 004767 002142 BPIER: JSR PC,SUPTAD ;SET UP PROG TBL ADR
001732' 042767 000010 176042 BIC #10,DFLGWD ;HOUSEKEEP ERROR FLAG


```

537 001740' 004567 001250          JSR      R5,ERRIS          ;REPORT INV BPI VALUE ERROR
538 001744' 002020                    .WORD    INVBPI-ERMBAS
539 001746' 000015                    .WORD    13.
540 001750' 000565                    BR       DERROR          ;GO TO ERROR EXIT
541
542 001752' 000 000          BPIVVL: .BYTE 00,000          ;VALID BPI VALUES
543 001754' 001 040          .BYTE 01,040
544 001756' 010 100          .BYTE 10,100
545 001760' 011 140          .BYTE 11,140
546 001762' 377 377          .BYTE 377,377
547
548
549          ;"READ" FUNCTION ROUTINE
550
551          ;JSR      R5,READ          FUNCTION CALL
552          ;.WORD    ADR          DATA ADDRESS (BITS 16 & 17)
553          ;.WORD    ADR          DATA ADDRESS (BITS 0 - 15)
554          ;.WORD    CNT          BYTE COUNT
555          ;.WORD    DEV          (NOT USED)
556
557 001764' 012701 000103          READ:  MOV      #103,R1          ;SET UP READ CMND CODE
558 001770' 012702 001421          MOV      #1421,R2          ;SET UP READ FLAG WORD
559 001774' 004767 001076          RDCOM: JSR      PC,CKDBSY          ;GO CK IF DEV IS BUSY
560 002000' 005267 177000          INC      RDCNT          ;ADD 1 TO READ CMND CNT
561 002004' 010700          MOV      PC,RO          ;SET UP ADR OF BYTES READ CNT
562 002006' 062700 176770          ADD      #BYRD+2-.,RO
563 002012' 000465          BR       CMDCOM          ;GO TO CMND COMMON PROCESSING
564
565          ;"WRITE" FUNCTION ROUTINE
566
567          ;JSR      R5,WRITE          FUNCTION CALL
568          ;.WORD    ADR          DATA ADDRESS (BITS 16 & 17)
569          ;.WORD    ADR          DATA ADDRESS (BITS 0 - 15)
570          ;.WORD    CNT          BYTE COUNT
571          ;.WORD    DEV          (NOT USED)
572
573 002014' 012701 000105          WRITE: MOV      #105,R1          ;SET UP WRITE CMND CODE
574 002020' 012702 001441          MOV      #1441,R2          ;SET UP CMND FLAG WORD
575 002024' 004767 001046          WRCOM: JSR      PC,CKDBSY          ;GO CK IF DEV IS BUSY
576 002030' 005267 176752          INC      WRCNT          ;ADD 1 TO WRITE CMND CNT
577 002034' 010700          MOV      PC,RO          ;SET UP ADR OF BYTES WRITTEN CNT
578 002036' 062700 176744          ADD      #BYWR+2-.,RO
579 002042' 000451          BR       CMDCOM          ;GO TO CMND COMMON PROCESSING
580
581          ;"WREIRG" FUNCTION ROUTINE
582
583          ;JSR      R5,WREIRG          FUNCTION CALL
584          ;.WORD    ADR          DATA ADDRESS (BITS 16 & 17)
585          ;.WORD    ADR          DATA ADDRESS (BITS 0 - 15)
586          ;.WORD    CNT          BYTE COUNT
587
588 002044' 012701 000115          WREIRG: MOV      #115,R1          ;SET UP WREIRG CMND CODE
589 002050' 012702 001442          MOV      #1442,R2          ;SET UP CMND FLAG WORD
590 002054' 000763          BR       WRCOM          ;GO TO COMMON WRITE PROCESSING

```



```

593                                     ;*WREOF" FUNCTION ROUTINE
594
595                                     ;JSR    R5,WREOF          FUNCTION CALL
596
597 002056' 012701 000107      WREOF:  MOV    #107,R1          ;SET UP WREOF CMND CODE
598 002062' 012702 000450      MOV    #450,R2          ;SET UP CMND FLAG WORD
599 002066' 000756      BR      WRCOM          ;GO TO COMMON WRITE PROCESSING
600
601
602                                     ;"SPFWD" FUNCTION ROUTINE
603
604                                     ;JSR    R5,SPFWD          FUNCTION CALL
605                                     ;.WORD  CNT          # OF RECORDS TO SPACE
606
607 002070' 012701 000111      SPFWD:  MOV    #111,R1          ;SET UP SPFWD CMND CODE
608 002074' 012702 003014      SPCOM:  MOV    #3014,R2        ;SET UP CMND FLAG WORD
609 002100' 004767 000772      MISCOM: JSR    PC,CKDBSY       ;GO CK IF DEV IS BUSY
610 002104' 005267 176700      INC    MISCNT          ;ADD 1 TO MISC. CMND CNT
611 002110' 000426      BR      CMDCOM          ;GO TO CMND COMMON PROCESSING
612
613
614                                     ;"SPREV" FUNCTION ROUTINE
615
616                                     ;JSR    R5,SPREV          FUNCTION CALL
617                                     ;.WORD  CNT          # OF RECORDS TO SPACE
618
619 002112' 012701 000113      SPREV:  MOV    #113,R1          ;SET UP SPREV CMND CODE
620 002116' 000766      BR      SPCOM          ;GO TO SPACE COM PROC.
621
622
623                                     ;"REWIND" FUNCTION ROUTINE
624
625                                     ;JSR    R5,REWIND          FUNCTION CALL
626
627 002120' 012701 000117      REWIND: MOV    #117,R1          ;SET UP REWIND CMND CODE
628 002124' 012702 011510      MOV    #11510,R2        ;SET UP CMND FLAG WORD
629 002130' 000763      BR      MISCOM          ;GO TO COMMON MISC. PROCESSING
630
631
632                                     ;"OFFLIN" FUNCTION ROUTINE
633
634                                     ;JSR    R5,OFFLIN          FUNCTION CALL
635
636 002132' 012701 000101      OFFLIN: MOV    #101,R1          ;SET UP OFFLIN CMND CODE
637 002136' 012702 000410      MOV    #410,R2          ;SET UP CMND FLAG WORD
638 002142' 000756      BR      MISCOM          ;GO TO MISC. CMND COM PROCESSING
639
640
641                                     ;"CRESET" FUNCTION ROUTINE
642
643                                     ;JSR    R5,CRESET          FUNCTION CALL
644
645 002144' 004767 000726      CRESET: JSR    PC,CKDBSY       ;GO CK IF DEV BUSY
646 002150' 005267 176634      INC    MISCNT          ;ADD 1 TO MISC. CMND CNT
647 002154' 004767 001714      JSR    PC,SUPTAD        ;SET UP MTC & PROG TBL ADR'S
648 002160' 052714 010000      BIS    #10000,(R4)      ;SET POWER CLEAR BIT IN MTC

```


649 002164' 000205 RTS R5 ;EXIT TO USER PROG

;COMMAND COMMON PROCESSING ROUTINE

;R4 = ADR OF MTC DEV REG
;R3 = PROG TBL ADR
;R2 = COMMAND FLAG WORD
;R1 = COMMAND CODE
;R0 = ADR OF BYTE COUNT TOTALS, IF APPLICABLE

;CMND FLAGWORD FORMAT:

:BIT 15 = ROLLBACK IN PROG (INT SRV.)
:BIT 14 = RC'VD 1ST RWD INT (INT SRV.)
:BIT 12 = T/O CNT BIT 4
:BIT 11 = T/O CNT BIT 3
:BIT 10 = T/O CNT BIT 2
:BIT 9 = T/O CNT BIT 1
:BIT 8 = T/O CNT BIT 0
:BIT 6 = REWIND CMND
:BIT 5 = DO WRITE ROLLBACK
:BIT 4 = DO READ ROLLBACK
:BIT 3 = DO NOT INCREMENT BYTE COUNTS
:BIT 2 = 1 ARGUMENT CMND
:BIT 1 = 3 ARGUMENT CMND
:BIT 0 = 4 ARGUMENT CMND

678	002166'	010067	176644	CMDCOM:	MOV	R0,CNTADR	;SAVE ADR OF BYTE COUNT
679	002172'	010267	176642		MOV	R2,CURFLG	;SAVE FLAGWD FOR TERMINATION
680	002176'	010267	176630		MOV	R2,TOEMAX	;SAVE # OF TIMEOUTS FOR CMND
681	002202'	005067	175614		CLR	ERR	;RESET THE ERROR INDICATOR
682	002206'	032702	000007		BIT	#7,R2	;THIS CMND HAVE ARGUMENTS?
683	002212'	001433			BEQ	10\$;Y,N-10\$
684	002214'	062704	000004		ADD	#4,R4	;POINT AT BUS ADR REG
685	002220'	032702	000004		BIT	#4,R2	;RECORD COUNT ONLY?
686	002224'	001011			BNE	5\$;N,Y-5\$
687	002226'	012500			MOV	(R5)+,R0	;GET BITS 16 & 17 OF BUS ADR
688	002230'	006300			ASL	R0	;ALIGN THEM TO CORRECT
689	002232'	006300			ASL	R0	;BIT POSITIONS
690	002234'	006300			ASL	R0	
691	002236'	006300			ASL	R0	
692	002240'	050001			BIS	R0,R1	;SET THEM INTO CMND CODE WORD
693	002242'	011514			MOV	(R5),(R4)	;GET BUS ADR BITS 0 - 15
694	002244'	012567	176600		MOV	(R5)+,RBADR	;SAVE ADR FOR ROLLBACK
695	002250'	012544		5\$:	MOV	(R5)+,-(R4)	;GET BYTE/RECORD COUNT
696	002252'	011467	176564		MOV	(R4),CURCNT	;SAVE THE COUNT
697	002256'	011467	176562		MOV	(R4),FINCNT	;INITIALIZE FINAL CNT TO SAME
698	002262'	005414			NEG	(R4)	;MAKE IT NEGATIVE
699	002264'	011467	176562		MOV	(R4),RBCNT	;SAVE CNT FOR ROLLBACK
700	002270'	005744			TST	-(R4)	;REALIGN REG ADR TO MTC
701	002272'	032702	000001		BIT	#1,R2	;4 ARGUMENT CMND?
702	002276'	001401			BEQ	10\$;Y,N-10\$
703	002300'	005725			TST	(R5)+	;BYPASS FOURTH ARGUMENT
704	002302'	116300	000035	10\$:	MOVB	PCURDV(R3),R0	;GET CURR DEV #

705	002306'	020027	000007		CMP	RO,#7	: INV DEV #?
706	002312'	101415			BLOS	DEVOK	: Y,N-DEVOK
707	002314'	004567	000656		JSR	R5,ERRCS	: GO REPORT INV DEV # ERROR
708	002320'	002006			.WORD	INVDVN-ERMBAS	
709	002322'	000012			.WORD	10.	
710	002324'	005267	176474		INC	DATAER	: ADD 1 TO DATA ERR CNT
711	002330'	012767	000001	175464	MOV	#1,ERR	: SET THE ERROR INDICATOR
712	002336'	005367	176460		DEC	ERRCNT	: REMOVE THE 1 ADDED TO DEV ERR CNT
713	002342'	000177	175502		JMP	3CUPGER	: GO TO MPG ERR RETN POINT
714	002346'	110064	000001		MOVB	RO,1(R4)	: PUT DEV # IN MTC BITS 9 THRU 10
715	002352'	016700	175424		MOV	DFLGWD,RO	: GET DEV ROUT FLGWD
716	002356'	042700	113777		BIC	#113777,RO	: RESET ALL BITS EXCEPT BPI & PARITY
717	002362'	050014			BIS	RO,(R4)	: SET UP BPI & PARITY
718	002364'	011467	176456		MOV	(R4),RBCMD	: SAVE DEV #, BPI & PARITY INFO
719	002370'	042767	000011	175404	BIC	#11,DFLGWD	: RESET THE ERROR FLAGS
720	002376'	005063	000030		CLR	PTOCNT(R3)	: INITIALIZE TIMEOUT COUNTER
721	002402'	005067	176422		CLR	TOECNT	: RESET # OF TIMEOUTS
722	002406'	005067	176442		CLR	NUMRB	: RESET # OF ROLLBACKS
723	002412'	005067	175372		CLR	EOF	: RESET EOF INDICATOR
724	002416'	005067	175370		CLR	EOT	: RESET EOT INDICATOR
725	002422'	052767	000002	175352	BIS	#2,DFLGWD	: SET THE "PROCESS TERMINATION" FLAG
726	002430'	052713	000010		BIS	#WT4IOT,(R3)	: SET WAITING FOR I/O TERM FLAG
727	002434'	110167	176406		MOVB	R1,RBCMD	: SAVE CMND CODE FOR ROLLBACK
728	002440'	110114			MOVB	R1,(R4)	: ISSUE THE CMND
729	002442'	005767	175334		TST	DFLGWD	: "NOWAIT" BIT SET?
730	002446'	100405			BMI	WTNOT	: N,Y-WTNOT
731	002450'	004577	175372		JSR	R5,3CIOBSY	: WAIT FOR I/O TO COMPLETE
732	002454'	004767	000730		JSR	PC,PROCTM	: GO PROCESS TERMINATION
733	002460'	000205			RTS	R5	: EXIT IN-LINE TO USER PROG
734							
735	002462'	042713	000010		WTNOT:	BIC	: RESET WAITING FOR I/O TERM
736	002466'	000774			BR	CMDEX	: GO TO EXIT


```

738                                     .SBTTL  TM11 INTERRUPT SERVICE ROUTINE
739
740
741 002470' 004000 001346          TMINT: JSR   R0, SAVREG      ;SAVE ALL REGISTERS
742 002474' 005267 176326          INC   INTCNT        ;ADD 1 TO INTERRUPT CNT
743 002500' 004767 001370          JSR   PC, SUPTAD    ;SET UP PROG TBL & MTC ADR'S
744 002504' 004567 001410          JSR   R5, STSTAT    ;STORE ALL DEV REG'S
745 002510' 176234          .WORD  ISTAT-
746 002512' 016702 176322          MOV   CURFLG, R2    ;GET THIS CMND'S FLGWD
747 002516' 005702          TST   R2            ;ROLLBACK IN PROGRESS?
748 002520' 100025          BPL   20$           ;Y, N-20$
749 002522' 042702 100000          BIC   #100000, R2   ;RESET ROLLBACK IN PROG FLG
750 002526' 016764 176316 000004  MOV   RBADR, 4(R4)  ;SET UP ORG DATA ADR
751 002534' 016764 176312 000002  MOV   RBCNT, 2(R4) ;SET UP ORG BYTE COUNT
752 002542' 016700 176300          MOV   RBCMD, R0     ;GET ORG CMND
753 002546' 042700 177761          BIC   #177761, R0  ;LEAVE ONLY THE FUNCT BITS
754 002552' 020027 000004          CMP   R0, #4        ;ORG CMD A WRITE DATA?
755 002556' 001003          BNE   10$           ;Y, N-10$
756 002560' 052767 000014 176260  BIS   #14, RBCMD    ;SET UP "WREIRG" CMND CODE
757 002566' 016714 176254 10$:  MOV   RBCMD, (R4)  ;RE-ISSUE CMND
758 002572' 000471          BR    90$           ;GO TO INT EXIT
759
760 002574' 005714          20$:  TST   (R4)         ;ERROR BIT SET?
761 002576' 100414          BMI   30$           ;N, Y-30$
762 002600' 032702 000100          25$:  BIT   #100, R2    ;THIS A REWIND CMND?
763 002604' 001455          BEQ   80$           ;Y, N-80$
764 002606' 032702 040000          BIT   #40000, R2   ;ALREADY RC'VD 1ST INT?
765 002612' 001052          BNE   80$           ;N, Y-80$
766 002614' 052702 040000          BIS   #40000, R2   ;SET RC'VD 1ST INT FLG
767 002620' 032744 000040          BIT   #40, -(R4)   ;AT BOT?
768 002624' 001044          BNE   70$           ;N, Y-70$
769 002626' 000453          BR    90$           ;GO TO INT EXIT
770
771                                     ;ERROR BIT SET
772
773 002630' 032744 040000          30$:  BIT   #40000, -(R4) ;EOF SET?
774 002634' 001407          BEQ   40$           ;Y, N-40$
775 002636' 012767 000001 175144  MOV   #1, EOF       ;SET THE EOF FLAG
776 002644' 005267 176146          INC   EOFcnt        ;ADD 1 TO EOF CNT
777 002650' 052702 000010          BIS   #10, R2      ;DON'T ADD IN ITS BYTE CNT
778 002654' 032714 002000          40$:  BIT   #2000, (R4) ;EOT SET?
779 002660' 001405          BEQ   45$           ;Y, N-45$
780 002662' 012767 000001 175122  MOV   #1, EOT       ;SET THE EOT FLAG
781 002670' 005267 176124          INC   EOTcnt        ;ADD 1 TO EOT CNT
782 002674' 032714 000100          45$:  BIT   #100, (R4)  ;SELECT REMOTE SET?
783 002700' 001413          BEQ   60$           ;Y, N-60$
784 002702' 032714 105600          BIT   #105600, (R4);ANY HARD ERRORS?
785 002706' 001010          BNE   60$           ;N, Y-60$
786 002710' 032714 030000          BIT   #30000, (R4);CRC OR LRC ERRORS?
787 002714' 001026          BNE   100$          ;N, Y-100$
788 002716' 005724          TST   (R4)+         ;POINT AT MTC
789 002720' 000727          BR    25$           ;GO CK CMND TYPE

```


E02

MAINDEC-11-DTTMA-B TM11/TU10 DEVICE ROUTINE FOR MPG
 DTTMAB.P11 TM11 INTERRUPT SERVICE ROUTINE

MACY11 27(732) 24-SEP-76 14:00 PAGE 6-1

SEQ 0405

```

791                                     ;ERROR & NORMAL TERMINATION
792
793 002722' 052767 000010 175052 50$: BIS #10,DFLGWD ;SET ROLLBACK EXH FLG
794 002730' 052767 000001 175044 60$: BIS #1,DFLGWD ;SET THE ERROR FLG
795 002736' 005724 70$: TST (R4)+ ;POINT R4 AT MTC
796 002740' 042714 000100 80$: BIC #100,(R4) ;RESET INT ENABLE
797 002744' 016467 000002 176072 MOV 2(R4),FINCNT ;STORE FINAL COUNT
798 002752' 042713 000010 BIC #WT4IOT,(R3) ;RESET WAITING FOR I/O TERM
799 002756' 010267 176056 90$: MOV R2,CURFLG ;STORE CMD FLGWD
800 002762' 004067 001070 JSR R0,RESREG ;RESTORE ALL REGISTERS
801 002766' 000177 175102 JMP @RTNINT ;EXIT FROM INTERRUPT
802
803
804                                     ;ROLLBACK TYPE OF ERROR
805
806 002772' 032702 000060 100$: BIT #60,R2 ;ROLLBACK TYPE OF CMND?
807 002776' 001754 BEQ 60$ ;Y,N-60$
808 003000' 032763 000400 000002 BIT #DOERCK,POPSW(R3) ;DO ERROR CK/RECOVERY?
809 003006' 001350 BNE 60$ ;Y,N-60$
810 003010' 032702 000040 BIT #40,R2 ;THIS A WRITE ROLLBACK?
811 003014' 001407 BEQ 110$ ;Y,N-110$
812 003016' 026767 176032 174762 CMP NUMRB,WRRB ;EXHAUSTED WR ROLLBACKS?
813 003024' 001736 BEQ 50$ ;N,Y-50$
814 003026' 005267 175762 INC WRBCNT ;INCR TOTAL # OF WR ROLLBACKS
815 003032' 000406 BR 120$ ;GO TO COMMON ROLLBACK PROC
816 003034' 026767 176014 174742 110$: CMP NUMRB,RDRB ;EXHAUSTED READ ROLLBACKS?
817 003042' 001727 BEQ 50$ ;N,Y-50$
818 003044' 005267 175742 INC RBCNT ;INCR TOTAL # OF READ ROLLBACKS
819 003050' 005267 176000 120$: INC NUMRB ;ADD 1 TO ROLLBACK CNT
820 003054' 052702 100000 BIS #100000,R2 ;SET ROLLBACK IN PROGRESS FLG
821 003060' 005724 TST (R4)+ ;POINT AT MTC
822 003062' 012764 177777 000002 MOV #-1,2(R4) ;SET UP RECORD CNT OF 1
823 003070' 112714 000113 MOVB #113,(R4) ;ISSUE SPACE REV CMND
824 003074' 000730 BR 90$ ;GO TO INT EXIT
  
```


.SBTTL SUBROUTINES FOR TM11 FUNCTION ROUTINES

;CHECK IF DEVICE IS BUSY AND WAIT IF IT IS

```
;JSR PC,CKDBSY      S/R CALL
;DESTROYS R0,R3,R4
;ON EXIT:
;R3 = PROG TBL ADR
;R4 = MTC ADR
```

```
826
827
828
829
830
831
832
833
834
835
836
837 003076' 004767 000772      CKDBSY: JSR    PC,SUPTAD      ;SET UP PROG TBL & MTC ADR'S
838 003102' 032714 000100      10$:   BIT    #100,(R4)  ;INT ENABLE ON?
839 003106' 001403              BEQ    20$           ;Y,N-20$
840 003110' 004577 174732      JSR    R5,DCIOBSY   ;RELEASE CONTROL
841 003114' 000772              BR     10$           ;GO CK AGAIN
842 003116' 032767 000002 174656 20$:   BIT    #2,DFLGWD    ;HAVE TO PROCESS PREV TERMINATION?
843 003124' 001403              BEQ    30$           ;Y,N-30$
844 003126' 004767 000256      JSR    PC,PROCTM    ;GO PROCESS TERMINATION
845 003132' 000763              BR     10$           ;GO RECHECK INT ENABLE
846 003134' 016767 174666 000012 30$:   MOV    IVCTAD,40$   ;STORE INT VECTOR ADR
847 003142' 016767 174662 000006      MOV    PSWD,45$    ;STORE PROG STATUS WORD
848 003150' 004577 174712      JSR    R5,SETVEC   ;GO SET UP THE VECTOR
849 003154' 000000              .WORD  XXXX        ;INT VECTOR ADR
850 003156' 000000              .WORD  XXXX        ;PSW
851 003160' 177310              .WORD  TMINT-      ;REL INT ROUT ADR
852 003162' 010567 175646      STSADR: MOV    R5,ERRADR ;SAVE CURR USER STMTN ADR
853 003166' 162767 000004 175640      SUB    #4,ERRADR
854 003174' 000207              RTS     PC
855
856
857
858
859
860
861
862
863
864
865
```

;ERROR INFORMATION DISPLAY S/R

```
;JSR R5,ERRCS      S/R CALL FOR CURR STATUS
;JSR R5,ERRIS      S/R CALL FOR INT STATUS
;.WORD MSGADR-ERMBAS REL ADR OF ERROR MSG
;.WORD MSGCNT      # OF BYTES IN ERROR MSG
;R3 = PROG TBL ADR
;DESTROYS R0,R1,R2
```

```
866 003176' 004567 000716      ERRCS: JSR    R5,STSTAT ;STORE CURR STATUS
867 003202' 175556              .WORD  CSTAT-      ;STORE ADR OF CURR STATUS
868 003204' 012767 175436 000110  ERRCS1: MOV    #CSTAT-ERSTAD,ERSTAD ;GO TO COMMON POINT
869 003212' 000403              BR     ERRCOM       ;STORE ADR OF LAST INT STATUS
870 003214' 012767 175422 000100  ERRIS: MOV    #ISTAT-ERSTAD,ERSTAD ;STORE MSG ADR
871 003222' 012567 000054      ERRCOM: MOV    (R5)+,ERMBAS ;STORE MSG CNT
872 003226' 012567 000052      MOV    (R5)+,ERMBAS+2 ;ADD 1 TO ERROR CNT
873 003232' 005267 175564      INC    ERRCNT      ;ERROR PRINTING INHIBITED?
874 003236' 032763 020000 000002  BIT    #PRONER,POPSW(R3) ;N,Y-ERREX
875 003244' 001060              BNE   ERREX        ;SAVE R4
876 003246' 010446              MOV    R4,-(SP)    ;SET USER MODE PRINT FLAG
877 003250' 005004              CLR    R4          ;DISPLAY UNIT #
878 003252' 004767 000670      JSR    PC,DISUNM   ;ROLLBACK EXHAUSTED?
879 003256' 032767 000010 174516  BIT    #10,DFLGWD  ;Y,N-5$
880 003264' 001404              BEQ    5$          ;ISSUE ROLLBK EXH MSG
881 003266' 004567 001072      JSR    R5,PRINT
```



```

882 003272' 002001 .WORD RBEXHM-.
883 003274' 000015 .WORD 13.
884 003276' 004567 001062 5$: JSR R5,PRINT ;PRINT ERROR MSG SPECIFIED
885 003302' 000000 ERMBAS: .WORD XXXX
886 003304' 000000 .WORD XXXX
887 003306' 026727 177770 002006 CMP ERMBAS,#INVDVN-ERMBAS ;INVALID UNIT # OR BPI ERROR?
888 003314' 103005 BHIS ERRSNM ;N,Y-ERRSNM
889 003316' 004567 000714 ERSTAD: JSR R5,DISPST ;DISPLAY STATUS REG'S
890 003322' 000000 .WORD XXXX
891 003324' 004767 000776 JSR PC,PRTIWD ;DISPLAY EOT & EOF VALUES
892 003330' 016300 000022 ERRSNM: MOV PSACST(R3),R0 ;GET ADR OF SRC STMENTS
893 003334' 111001 10$: MOV (R0),R1 ;SAVE STMT LENGTH
894 003336' 026067 000004 175470 CMP 4(R0),ERRADR ;ERROR OCCUR ON THIS STMT?
895 003344' 001402 BEQ 20$ ;N,Y-20$
896 003346' 060100 ADD R1,R0 ;POINT AT NXT STMT
897 003350' 000771 BR 10$ ;GO CK NXT STMT
898 003352' 005720 20$: TST (R0)+ ;SET UP ADR OF STMT # DATA
899 003354' 010701 MOV PC,R1 ;SET UP DATA OUTPUT ADR
900 003356' 062701 001646 ADD #STNUM-. ,R1
901 003362' 004577 174474 JSR R5,DECA$C ;CONVERT IT TO ASCII
902 003366' 012767 020040 001634 MOV #20040,STNUM+4 ;SET 2 LOW DIGITS TO SPACES
903 003374' 004567 000764 JSR R5,PRINT ;ISSUE STMT # MSG
904 003400' 001614 .WORD STNMNG-.
905 003402' 177762 .WORD -14.
906 003404' 012604 ERREX: MOV (SP)+,R4 ;RESTORE R4
907 003406' 000205 RTS R5 ;EXIT IN-LINE
908
909
910 ;PROCESS TERMINATION OF PREVIOUS I/O FUNCTION
911
912 ;JSR PC,PROCTM S/R CALL
913
914 003410' 004067 000426 PROCTM: JSR R0,SAVREG ;SAVE ALL REG'S
915 003414' 042767 000002 174360 BIC #2,DFLGWD ;RESET PROCESS TERMINATION FLAG
916 003422' 032767 000010 175410 BIT #10,CURFLG ;INCR BYTE COUNT?
917 003430' 001015 BNE 6$ ;Y,N-6$
918 003432' 016700 175404 MOV CURCNT,R0 ;GET INITIAL BYTE CNT
919 003436' 016701 175402 MOV FINCNT,R1 ;GET FINAL BYTE CNT
920 003442' 100001 BPL 2$ ;IS IT NEGATIVE? (Y,N-2$)
921 003444' 005401 NEG R1 ;MAKE IT POSITIVE
922 003446' 160100 2$: SUB R1,R0 ;SUB REMAINING CNT FROM INITIAL CNT
923 003450' 010067 174344 MOV R0,SIZE ;STORE # OF BYTES ACTUALLY XFERRED
924 003454' 016701 175356 MOV CNTADR,R1 ;GET ADR OF BYTE CNT TOTALS
925 003460' 060011 ADD R0,(R1) ;ADD IN THIS CNT
926 003462' 005541 ADC -(R1) ;UPDATE MOST SIGNF WORD OF CNT
927 003464' 032767 000001 174310 6$: BIT #1,DFLGWD ;WAS THERE AN ERROR?
928 003472' 001504 BEQ 80$ ;Y,N-80$
929 003474' 012767 000001 174320 MOV #1,ERR ;SET THE ERROR INDICATOR
930 003502' 032763 000400 000002 BIT #DOERCK,POPSW(R3) ;SUPPOSED TO DO ERROR CHECKING?
931 003510' 001073 BNE 70$ ;Y,N-70$
932 003512' 010701 MOV PC,R1 ;GET ADR OF CODE AREA IN ERR MSG
933 003514' 062701 001534 ADD #CODFLD-. ,R1
934 003520' 010102 MOV R1,R2 ;MOVE IT TO WORK REG
935 003522' 012700 000023 MOV #19,R0 ;SET UP AREA SIZE
936 003526' 112722 000040 10$: MOVB #40,(R2)+ ;CLEAR AREA TO SPACES
937 003532' 005300 DEC R0
    
```



```

938 003534' 001374      BNE      10$
939 003536' 010700      MOV      PC,R0          ;SET UP ADR OF ERROR CODE TBL
940 003540' 062700 000156  ADD      #ERCDTB-.,R0
941 003544' 010702      MOV      PC,R2          ;SET UP ADR OF STORED DEV REG'S - 1
942 003546' 062702 175175  ADD      #I$STAT-1-.,R2
943 003552' 005046      CLR      -(SP)
944 003554' 005202      15$: INC      R2          ;INITIALIZE CODE CNT
945 003556' 112004      20$: MOVB   (R0)+,R4      ;POINT AT NXT STATUS BYTE
946 003560' 120427 000377  CMPB   R4,#377        ;GET ERROR BIT MASK CODE
947 003564' 001773      BEQ     15$           ;GO TO NXT STAT BYTE CODE?
948 003566' 005704      TST    R4            ;N,Y-15$
949 003570' 001427      BEQ     60$          ;END OF THE CODE TBL?
950 003572' 120427 000376  CMPB   R4,#376        ;N,Y-60$
951 003576' 001004      BNE     30$          ;BIT VALUE OF 0 = ERR COND?
952 003600' 112004      MOVB   (R0)+,R4      ;Y,N-30$
953 003602' 130412      BITB   R4,(R2)       ;GET BIT VALUE
954 003604' 001406      BEQ     40$          ;THIS BIT RESET IN STAT BYTE?
955 003606' 000402      BR     35$          ;N,Y-40$
956 003610' 130412      30$: BITB   R4,(R2)   ;GO TO NXT TBL ENTRY
957 003612' 001003      BNE     40$          ;THIS ERROR BIT SET IN STATUS BYTE?
958 003614' 062700 000003  35$: ADD     #3,R0      ;N,Y-40$
959 003620' 000756      BR     20$          ;POINT AT NXT CODE TBL ENTRY
960 003622' 005716      40$: TST    (SP)       ;GO CK FOR NXT CODE
961 003624' 001402      BEQ     50$          ;FIRST ERROR CODE IN MSG?
962 003626' 112721 000054  50$: MOVB   #' ,(R1)+   ;N,Y-50$
963 003632' 005216      INC     (SP)         ;MOVE COMMA TO MSG
964 003634' 112021      MOVB   (R0)+,(R1)+   ;INC # OF CODES IN THE MSG
965 003636' 112021      MOVB   (R0)+,(R1)+   ;MOVE ERROR CODE TO MSG
966 003640' 112021      MOVB   (R0)+,(R1)+
967 003642' 022716 000005  CMP     #5,(SP)
968 003646' 001343      BNE     20$          ;PUT 5 CODES IN THE MSG?
969 003650' 005726      60$: TST    (SP)+     ;Y,N-20$
970 003652' 004567 177336  JSR    R5,ERRIS      ;RESTORE STACK
971 003656' 001730      .WORD  TM$MSG-ERMBAS ;GO ISSUE STATUS ERROR MSG
972 003660' 000041      .WORD  33
973 003662' 004767 000100  65$: JSR    PC,RINTV    ;GO RESET INT VECTOR
974 003666' 004067 000164  JSR    R0,RESREG     ;RESTORE REG'S
975 003672' 004577 174152  JSR    R5,@CUPGER    ;GO TO MPG ERR RETN POINT
976 003676' 000207      RTS     PC           ;EXIT IN-LINE
977 003700' 005267 175116  70$: INC     ERRCNT    ;ADD 1 TO ERROR CNT
978 003704' 004767 000056  80$: JSR    PC,RINTV    ;GO RESET INT VECTOR
979 003710' 004067 000142  JSR    R0,RESREG     ;RESTORE REG'S
980 003714' 000207      RTS     PC           ;EXIT IN-LINE
981
982
983 003716' 047200 046530      ERCDTB: .ASCII <200>/NXM/ ;ERROR MSG CODE TABLE
984 003722'      376      .BYTE  376
985 003723'      100 046123      122 .ASCII <100>/SLR/
986 003727'      004 051127      114 .ASCII <004>/WRL/
987 003733'      377      .BYTE  377
988 003734' 044600 041514      .ASCII <200>/ILC/
989 003740' 041440 042522      .ASCII <040>/CRE/
990 003744' 050020 042501      .ASCII <020>/PAE/
991 003750' 041010 046107      .ASCII <010>/BGL/
992 003754' 051002 042514      .ASCII <002>/RLE/
993 003760' 041001 042524      .ASCII <001>/BTE/

```



```

994 003764' 000 .BYTE 0 ;TABLE TERMINATOR
995 003766' .EVEN
996
997
998 ;RESET INTERRUPT VECTOR S/R
999
1000 ;JSR PC,RINTV S/R CALL
1001 ;R3 MUST CONTAIN PROG TBL ADR
1002 ;DESTROYS R0
1003
1004 003766' 004567 000020 RINTV: JSR R5,TVECT ;GO CK IF I HAVE VECTOR CONTROL
1005 003772' 000406 BR RINTEX ;BR IF I DON'T
1006 003774' 016767 174026 000004 MOV IVCTAD,10$ ;GET CURR INT VECT ADR
1007 004002' 004577 174062 JSR R5,@CLAVEC ;GO HAVE MPG CLEAR IT
1008 004006' 000000 10$: .WORD XXXX
1009 004010' 000207 RINTEX: RTS PC ;EXIT IN-LINE
1010
1011
1012 ;TEST INTERRUPT VECTOR S/R
1013
1014 ;JSR R5,TVECT S/R CALL
1015 ;BR LABEL EXECUTED IF NOT SAME
1016 ;R3 MUST CONTAIN PROG TBL ADR
1017 ;DESTROYS R0
1018
1019 004012' 016767 174010 000010 TVECT: MOV IVCTAD,20$ ;GET CURR INT VECT ADR
1020 004020' 016346 000004 MOV PFWADR(R3),-(SP) ;STORE FLGWD ADR TO IDENTIFY ME
1021 004024' 004577 174042 JSR R5,@TSTVEC ;DO I HAVE VECTOR CONTROL?
1022 004030' 000000 20$: .WORD XXXX ; MPG WILL TELL ME SINCE I CAN'T
1023 004032' 176436 .WORD TMINT- ; GET AT LOWER MEM IF MEM MGMNT
1024 004034' 000401 BR TVECTX ;BR IF I DONT'T HAVE CNTRL
1025 004036' 005725 TST (R5)+ ;BYPASS BR INST IN S/R CALL
1026 004040' 000205 TVECTX: RTS R5 ;EXIT IN-LINE
    
```



```

1028                .SBTTL  SUBROUTINES FOR TM11 DEVICE ROUTINE
1029
1030
1031                ;SAVE REGISTERS R0 THRU R5
1032
1033                ;JSR    RO,SAVREG          S/R CALL
1034
1035                SAVREG: MOV    R1,-(SP)      ;SAVE R0 THRU R5
1036                MOV    R2,-(SP)
1037                MOV    R3,-(SP)
1038                MOV    R4,-(SP)
1039                MOV    R5,-(SP)
1040                MOV    RO,PC              ;EXIT IN-LINE
1041
1042
1043                ;RESTORE REGISTERS R0 THRU R5
1044
1045                ;JSR    RO,RESREG        S/R CALL
1046
1047                RESREG: TST    (SP)+      ;RESTORE R5 THRU R0
1048                MOV    (SP)+,R5
1049                MOV    (SP)+,R4
1050                MOV    (SP)+,R3
1051                MOV    (SP)+,R2
1052                MOV    (SP)+,R1
1053                RTS    RO                ;EXIT IN-LINE
1054
1055
1056                ;SET PROGRAM'S PROG TABLE ADR IN R3 & MTC ADR IN R4
1057
1058                ;JSR    PC,SUPTAD        S/R CALL
1059
1060                SUPTAD: MOV    PC,R3      ;SET UP LOCATION ZERO ADR
1061                ADD    #LOCZ-,R3
1062                SUB    -2(R3),R3        ;SUBTRACT PROG TBL LENGTH
1063                MOV    DREGAD,R4        ;GET DEV REG BASE ADR
1064                ADD    #2,R4           ;POINT AT MTC
1065                RTS    PC              ;EXIT IN-LINE
1066
1067
1068                ;STORE DEVICE'S STATUS REGISTERS
1069
1070                ;JSR    R5,STSTAT        S/R CALL
1071                ;.WORD  STADR-          REL STORAGE ADR
1072                ;DESTROYS R0,R1
1073
1074                STSTAT: MOV    R5,R1      ;GET REL STORAGE ADR & MAKE
1075                ADD    (R5)+,R1        ;IT ABSOLUTE
1076                MOV    DREGAD,RO        ;GET ADR OF DEV REG'S
1077                MOV    (RO)+,(R1)+     ;STORE ALL DEV REG'S
1078                MOV    (RO)+,(R1)+
1079                MOV    (RO)+,(R1)+
1080                MOV    (RO)+,(R1)+
1081                MOV    (RO)+,(R1)+
1082                MOV    (RO),(R1)
1083                RTS    R5              ;EXIT IN-LINE
    
```



```

1084
1085
1086                ;DISPLAY CURRENT UNIT #
1087
1088                ;JSR    PC,DISUNM      S/R CALL
1089                ;R3 MUST CONTAIN PROG TBL ADR
1090                ;DESTROYS R0,R1,R2
1091
1092 004146' 012767 000026 000056 DISUNM: MOV    #22.,DISUML      ;INITIALIZE TO NORMAL MSG LNGTH
1093 004154' 116300 000035          MOVB   PCURDV(R3),R0    ;GET CURR UNIT #
1094 004160' 020027 000007          CMP    R0,#7        ;VALID UNIT #?
1095 004164' 101007          BHI    DISUIV      ;Y,N-DISUIV
1096 004166' 004577 173666          JSR    R5,@BTASLZ  ;CONVERT # TO DECIMAL ASCII
1097 004172' 000402          .WORD  UNASCII-
1098 004174' 016767 000400 000372  MOV    UNASCII+4,UNASCII ;MOVE ASCII # TO 1ST TWO DIGITS
1099 004202' 000410          BR     DISUPR      ;GO ISSUE MSG
1100 004204' 012767 000032 000020 DISUIV: MOV    #26.,DISUML      ;SET UP ERR COND MSG LNGTH
1101 004212' 042700 177400          BIC    #177400,R0  ;RESET HIGH BYTE
1102 004216' 004577 173634          JSR    R5,@BINASC ;CONVERT BINARY # TO ASCII
1103 004222' 000352          .WORD  UNASCII-
1104 004224' 004567 000134          DISUPR: JSR   R5,PRINT ;GO ISSUE UNIT # MSG
1105 004230' 000320          .WORD  UNITMG-
1106 004232' 000026          DISUML: .WORD  22.
1107 004234' 000207          RTS     PC        ;EXIT IN-LINE
1108
1109
1110                ;TAILOR STATUS MSG & PRINT IT
1111
1112                ;JSR    R5,DISPST     S/R CALL
1113                ;WORD  STATADR-      REL ADR OF STATUS DATA
1114                ;DESTROYS R0,R1,R2
1115
1116 004236' 010502          DISPST: MOV   R5,R2      ;GET REL DATA ADR
1117 004240' 062502          ADD    (R5)+,R2     ;MAKE IT ABS
1118 004242' 010701          MOV    PC,R1        ;SET UP ADR OF REG NAMES IN ASCII
1119 004244' 062701 173652          ADD    #DVREGS-,R1
1120 004250' 012746 000006          MOV    #DVREGE-DVREGS/6,-(SP) ;GET # OF REGISTERS TO DISPLAY
1121 004254' 012167 000322          10$:  MOV    (R1)+,DVRGMG ;MOVE REG NAME TO MSG
1122 004260' 012167 000320          MOV    (R1)+,DVRGMG+2
1123 004264' 005721          TST    (R1)+        ;BYPASS DISP VALUE
1124 004266' 012200          MOV    (R2)+,R0     ;GET REG'S STORED VALUE
1125 004270' 010146          MOV    R1,-(SP)     ;SAVE R1 & R2
1126 004272' 010246          MOV    R2,-(SP)
1127 004274' 004577 173556          JSR    R5,@BINASC  ;CONVERT IT TO ASCII
1128 004300' 000310          .WORD  DVRGDT-
1129 004302' 004567 000056          JSR    R5,PRINT    ;PRINT THE STATUS MSG
1130 004306' 000274          .WORD  DVRGMG-
1131 004310' 000014          .WORD  12.
1132 004312' 012602          MOV    (SP)+,R2     ;RESTORE R1 & R2
1133 004314' 012601          MOV    (SP)+,R1
1134 004316' 005316          DEC    (SP)         ;DECR REG CNT
1135 004320' 001355          BNE    10$         ;DONE ALL? (Y,N-10$)
1136 004322' 005726          TST    (SP)+        ;REMOVE COUNT FROM STACK
1137 004324' 000205          RTS     R5        ;EXIT IN-LINE

```



```

1139                                     ;DISPLAY CONTENTS OF EOF & EOT WORDS
1140
1141                                     ;JSR   PC,PRTIWD       S/R CALL
1142                                     ;DESTROYS R0,R1,R2
1143
1144 004326' 016700 173456   PRTIWD: MOV   EOF,R0           ;GET EOF VALUE
1145 004332' 004577 173520   JSR   R5,@BINASC       ;CONVERT ITS VALUE TO ASCII
1146 004336' 000633           .WORD  IFEOF-
1147 004340' 016700 173446   MOV   EOT,R0           ;GET EOT VALUE
1148 004344' 004577 173506   JSR   R5,@BINASC       ;CONVERT IT TO ASCII
1149 004350' 000635           .WORD  IFEOT-
1150 004352' 004567 000006   JSR   R5,PRINT         ;PRINT MSG WITH THEIR VALUES
1151 004356' 000606           .WORD  INFOMG-
1152 004360' 000027           .WORD  23.
1153 004362' 000207           RTS    PC               ;EXIT IN-LINE
1154
1155                                     ;ISSUE MSG TO LIST DEVICE
1156
1157                                     ;JSR   R5,PRINT       S/R CALL
1158                                     ;.WORD MSGADR-      REL ADR OF MSG
1159                                     ;.WORD BYTCNT       MSG BYTE CNT (IF NEGATIVE,
1160                                     ;                   RESET PRT DEV DEDICATED.)
1161                                     ;R3 = PROG TBL ADR
1162                                     ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
1163                                     ;DESTROYS R0,R1,R2
1164
1165 004364' 010500           PRINT: MOV   R5,R0           ;GET MSG ADR & MAKE IT ABS
1166 004366' 062500           ADD   (R5)+,R0
1167 004370' 012501           MOV   (R5)+,R1         ;GET BYTE COUNT
1168 004372' 005704           TST   R4               ;USE CMND MODE PRINT?
1169 004374' 100030           BPL   40$              ;Y,N-40$
1170 004376' 010702           MOV   PC,R2           ;SET UP LINK INFO ADR
1171 004400' 062702 000040   ADD   #20$-,R2
1172 004404' 160200           SUB   R2,R0           ;MAKE MSG ADR REL
1173 004406' 010022           MOV   R0,(R2)+        ;STORE MSG ADR
1174 004410' 010112           MOV   R1,(R2)         ;STORE MSG'S BYTE COUNT
1175 004412' 100001           BPL   10$              ;CNT NEG? (Y,N-10$)
1176 004414' 005412           NEG   (R2)            ;MAKE IT POSITIVE
1177 004416' 016367 000006 000056 10$: MOV   PASCIN(R3),PROGNM ;STORE PROG'S # IN MSG
1178 004424' 004577 173424   JSR   R5,@CLIST        ;ISSUE PROG #
1179 004430' 000050           .WORD  PNMMSG-
1180 004432' 000005           .WORD  5
1181 004434' 004577 173414   JSR   R5,@CLIST        ;ISSUE MSG SPECIFIED
1182 004440' 000000           .WORD  XXXX
1183 004442' 000000           .WORD  XXXX
1184 004444' 004577 173404   JSR   R5,@CLIST        ;ISSUE A <CR> & <LF>
1185 004450' 000220           .WORD  CRLF-
1186 004452' 000002           .WORD  2
1187 004454' 000410           BR    PRTEX           ;GO TO EXIT
1188 004456' 010067 000010 40$: MOV   R0,50$          ;STORE MSG'S ABS ADR
1189 004462' 010167 000006   MOV   R1,60$          ;STORE ITS BYTE CNT
1190 004466' 004577 173360   JSR   R5,@JLIST        ;GO TO MPG TO ISSUE THE MSG
1191 004472' 000000           .WORD  XXXX
1192 004474' 000000           .WORD  XXXX
1193 004476' 000205   PRTEX: RTS    R5       ;EXIT IN-LINE
    
```



```

1195 .SBTTL TM11 MESSAGE STORAGE AREA
1196
1197
1198 .NLIST BEX
1199
1200 .EVEN
1201 004500' 021520 PNMMSG: .ASCII /P#/
1202 004502' 054130 011 PROGM: .ASCII /XX/<011>
1203 004505' 101 020124 040514 ATIMSG: .ASCII 'AT LAST INT:'
1204 004521' 103 051125 042522 CURMSG: .ASCII /CURRENTLY:/
1205 004533' 105 042116 047440 RENDMG: .ASCII /END OF REPORT/
1206 .EVEN
1207 004550' 025052 025052 046524 UNITMG: .ASCII /***TM11 TAPE UNIT: /
1208 004574' 054130 054130 054130 UNASCI: .ASCII /XXXXXX/
1209 .EVEN
1210 004602' 054130 054130 020075 DVRCMG: .ASCII /XXXX= /
1211 004610' 054130 054130 054130 DVRGDT: .ASCII /XXXXXX/
1212 004616' 054502 042524 035123 CNTSMG: .ASCII /BYTES: RD= /
1213 004632' 054130 054130 054130 BCMRD: .ASCII /XXXXXXXXXXXXX WR= /
1214 004654' 054130 054130 054130 BCMWR: .ASCII /XXXXXXXXXXXXX/
1215 004670' 005015 041411 047115 CRLF: .ASCII <015><012><011>/CMNDS: RD= /
1216 004707' 130 054130 054130 CMDCRD: .ASCII /XXXXXX WR= /
1217 004722' 054130 054130 054130 CMDCWR: .ASCII /XXXXXX MISC= /
1218 004737' 130 054130 054130 CMDCMS: .ASCII /XXXXXX/<015><012><011>/ROLLBACKS: RD= /
1219 004770' 054130 054130 054130 CNTRRB: .ASCII /XXXXXX WR= /
1220 005003' 130 054130 054130 CNTWRB: .ASCII /XXXXXX/<015><012><011>/# OF EOF'S= /
1221 005030' 054130 054130 054130 CNTEOF: .ASCII /XXXXXX EOT'S= /
1222 005046' 054130 054130 054130 CNTEOT: .ASCII /XXXXXX/<015><012><011>/ERRORS: DEV= /
1223 005075' 130 054130 054130 CNTERR: .ASCII /XXXXXX DATA= /
1224 005112' 054130 054130 054130 CNTDER: .ASCII /XXXXXX/<015><012><011>/INTERRUPTS: /
1225 005140' 054130 054130 054130 CNTINT: .ASCII /XXXXXX/
1226 005146' 005146' CNTSEN=
1227 005146' 044524 042515 052517 IOTO: .ASCII 'TIMEOUT ON I/O'
1228 005164' 047505 036506 040 INFOMG: .ASCII /EOF= /
1229 005171' 130 054130 054130 IFEOF: .ASCII /XXXXXX EOT= /
1230 005205' 130 054130 054130 IFEOT: .ASCII /XXXXXX/
1231 .EVEN
1232 005214' 052123 047115 020124 STMNMG: .ASCII /STMNT # /
1233 005224' 054130 054130 054130 STMNUM: .ASCII /XXXXXX/
1234 005232' 052123 052101 051525 TMEMSG: .ASCII /STATUS ERROR: /
1235 005250' 000023 CODFLD: .BLKB 19.
1236
1237 005273' 122 046117 041114 RBEXHM: .ASCII /ROLLBACK EXH./
1238 005310' 047111 020126 047125 INVAVN: .ASCII /INV UNIT #/
1239 005322' 047111 020126 050102 INVBPI: .ASCII /INV BPI VALUE/
1240 .EVEN
1241
1242 .LIST BEX
1243
1244 005340' DVREND=

```



```

1246          .SBTTL FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES
1247
1248          ; PROGRAM TABLE FORMAT
1249
1250          000242 PTLGTH= 162. ;PROGRAM TABLE LENGTH - NON MEM MGMNT VERSION OF MPG
1251
1252          ;(PTLGTH= 212. ;PROGRAM TABLE LENGTH - MEM MGMNT VERSION OF MPG)
1253
1254          000000 PFLGWD= +0. ;PROGRAM FLAG WORD - 1 WORD
1255
1256          000002 URSTOP= 2 ; 1 = USER HAS STOPPED THIS PROGRAM
1257          000004 ERSTOP= 4 ; 1 = AN ERROR HAS STOPPED THIS PROGRAM
1258          000010 WT4IOT= 10 ; 1 = WAITING FOR I/O TERMINATION
1259          000020 CTPRIO= 20 ; 1 = CONSOLE OR PRINTER I/O IN PROGRESS
1260          000040 SETDED= 40 ; 1 = THIS PROG SET THE PRT DEV DEDICATED FLAG
1261          000100 OCPRES= 100 ; 1 = OBJ CODE IS PRESENT
1262          000200 USEUBM= 200 ; 1 = THIS PROG USES THE UNIBUS MAP (MEM MGMNT ONLY)
1263          100000 ACTIVE= 100000 ; 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)
1264
1265          000002 POPSW= +2. ;PROGRAM'S OPERATION SWITCHES - 1 WORD
1266
1267          100000 STONER= 100000 ; 1 = STOP PROG EXECUTION UPON ERROR
1268          040000 CYCPRG= 40000 ; 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
1269          020000 PRONER= 20000 ; 1 = DO NOT PRINT ON ERROR
1270          010000 BIT12= 10000 ; 0 = NOT USED
1271          004000 BIT11= 4000 ; 0 = NOT USED
1272          002000 CYCDVL= 2000 ; 1 = CYCLE THE DEVICE LIST
1273          001000 GTNXTD= 1000 ; 1 = CYCLE ON SAME DEVICE UPON ERROR
1274          000400 DOERCK= 400 ; 1 = DON'T DO ERROR CHECKING
1275          000200 SPOPER= 200 ; 1 = DEVICE SPECIAL OPERATION
1276          000100 BIT6= 100 ; 0 = NOT USED
1277          000040 DOIOT= 40 ; 1 = DO NOT PERFORM I/O TIMEOUT
1278          000020 AUTORP= 20 ; 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
1279          000010 AURPEP= 10 ; 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
1280          000004 HSKPEP= 4 ; 1 = HOUSEKEEP COUNTS ONLY AT RUN COMMAND
1281          000002 PFBBOV= 2 ; 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
1282          000001 NOCOMP= 1 ; 1 = DO NOT PRINT PROG COMPLETED MSG
1283
1284          000004 PFWADR= +4. ;*;PROGRAM FLAGWORD ADDRESS - 1 WORD
1285
1286          000006 PASCIN= +6. ;PROGRAM'S NUMBER IN ASCII - 1 WORD
1287
1288          000010 PNAME= +8. ;PROGRAM'S NAME IN ASCII - 6 BYTES
1289
1290          000016 PRDIOA= +14. ;ADDRESS OF READ I/O AREA - 1 WORD
1291
1292          000020 PWRIOA= +16. ;ADDRESS OF WRITE I/O AREA - 1 WORD
1293
1294          000022 PSRCST= +18. ;SOURCE STATEMENTS START ADDRESS - 1 WORD
1295
1296          000024 POBJST= +20. ;OBJECT CODE START ADDRESS - 1 WORD
1297
1298          000026 PLNGTH= +22. ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD
1299
1300          000030 PTOCNT= +24. ;I/O TIMEOUT COUNT - 1 WORD
1301
    
```


1302	000032	PMDLCD= +26.	;DEV ROUT MODEL # CODE - 1 WORD
1303	000034	PDPNTR= +28.	;CURRENT DEVICE NUMBER POINTER - 1 BYTE
1304	000035	PCURDV= +29.	;CURRENT DEVICE # - 1 BYTE
1305	000036	PDNUMS= +30.	;DEVICE NUMBERS - 16 BYTES
1306	000056	PTEM0= +46.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1307	000060	PTEM1= +48.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1308	000062	PTEM2= +50.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1309	000064	PTEM3= +52.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1310	000066	PTEM4= +54.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1311	000070	PTEM5= +56.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1312	000072	PTEM6= +58.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1313	000074	PTEM7= +60.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1314	000076	PTEM8= +62.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1315	000100	PTEM9= +64.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1316	000102	PTEM10= +66.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1317	000104	PTEM11= +68.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1318	000106	PTEM12= +70.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1319	000110	PTEM13= +72.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1320	000112	PTEM14= +74.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1321	000114	PTEM15= +76.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1322	000116	PNBR= +78.	;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
1323	000120	PSRC= +80.	;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
1324	000122	PDST= +82.	;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
1325	000124	PSTKCT= +84.	;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
1326	000126	PSTKSV= +86.	;STACK WORDS STORAGE AREA - 30 WORDS
1327	000222	PSVREG= +146.	;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
1328	000236	PUSRPC= +158.	;USER'S CURRENT PROGRAM COUNTER - 1 WORD


```

1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375      000240
1376
1377
1378
1379      000242
1380
1381

```

;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMNT VERSION
;(PRDIOX= +160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
;(PRDIOV= +164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
;(PWRIOX= +168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
;(PWRIOV= +172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
;(PUPARS= +176. ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)
;(PUPDRS= +192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
;(PUBMAP= +208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)
;END OF MEM MGMNT ONLY ENTRIES
PTSIZE= +160. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMNT
;(PTSIZE= +210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMNT VERSION)
PTEND= +162. ;END OF PROGRAM TABLE - NON MEM MGMNT VERSION
;(PTEND= +212. ;END OF PROGRAM TABLE - MEM MGMNT VERSION)


```

1383           ;      DEVICE ROUTINE TABLE
1384
1385           000116      DRTLTH= 78.      ;DEVICE ROUTINE TABLE LENGTH
1386
1387           ;
1388           000000      DEVRSZ= +0.      ;DEVICE ROUTINE SIZE IN BYTES - 1 WORD
1389
1390           000002      DEVFWD= +2.      ;DEVICE ROUTINE FLAGWORD - 1 WORD
1391
1392           000004      DEVIW1= +4.      ;DEVICE INTERFACE WORD # 1 - 1 WORD
1393
1394           000006      DEVIW2= +6.      ;DEVICE INTERFACE WORD # 2 - 1 WORD
1395
1396           000010      DEVIW3= +8.      ;DEVICE INTERFACE WORD # 3 - 1 WORD
1397
1398           000012      DEVIW4= +10.     ;DEVICE INTERFACE WORD # 4 - 1 WORD
1399
1400           000014      DEVIW5= +12.     ;DEVICE INTERFACE WORD # 5 - 1 WORD
1401
1402           000016      DEVIW6= +14.     ;DEVICE INTERFACE WORD # 6 - 1 WORD
1403
1404           000020      DEVIW7= +16.     ;DEVICE INTERFACE WORD # 7 - 1 WORD (SIZE)
1405
1406           000022      DEVIW8= +18.     ;DEVICE INTERFACE WORD # 8 - 1 WORD (ERR)
1407
1408           000024      DEVDRA= +20.     ;DEVICE REGISTERS ADDRESS - 1 WORD
1409
1410           000026      DEVIVA= +22.     ;DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
1411
1412           000030      DEVRPS= +24.     ;DEVICE READ PROCESSOR STATUS WORD (BUS REQ) - 1 WORD
1413
1414           000032      DEVPWS= +26.     ;DEVICE WRITE PROC STATUS WORD (BUS REQ) - 1 WORD
1415
1416           000034      DHKPAD= +28.     ;DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
1417
1418           000036      DERPAD= +30.     ;DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
1419
1420           000040      DKILAD= +32.     ;DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
1421
1422           000042      DECTAD= +34.     ;DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
1423
1424           000044      DTOEAD= +36.     ;DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
1425
1426           000046      DEVI0B= +38.     ;DEVICE I/O BUSY BRANCH ADDRESS (CIOBSY) - 1 WORD
1427
1428           000050      DEVDER= +40.     ;DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
1429
1430           000052      DVUPRT= +42.     ;USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
1431
1432           000054      DVCPRN= +44.     ;CMND MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
1433
1434           000056      DEVBTA= +46.     ;CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
1435
1436           000060      DVBTDA= +48.     ;CONVERT BINARY TO DECIMAL ASCII BR ADR (BTASLZ) - 1 WORD
1437
1438

```


E03

MAINDEC-11-DTTMA-B TM11/TU10 DEVICE ROUTINE FOR MPG MACY11 27(732) 24-SEP-76 14:00 PAGE 10-4
 DTTMAB.P11 FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

SEQ 0418

1439	000062	DVDPDA= +50.	; CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
1440			
1441	000064	DVSFWD= +52.	; MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
1442			
1443	000066	DVSVEC= +54.	; SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
1444			
1445	000070	DVCVEC= +56.	; CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
1446			
1447	000072	DVTVEC= +58.	; TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
1448			
1449	000074	DVRINT= +60.	; RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
1450			
1451	000076	DVGETB= +62.	; GET DATA BYTE BR ADR (GETBYT) - 1 WORD
1452			
1453	000100	DVPUTB= +64.	; PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
1454			
1455	000102	DEVSTP= +66.	; DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
1456			
1457	000104	DEVETP= +68.	; DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
1458			
1459	000106	DVPTEP= +70.	; PACK TABLE EXTEN. REL POINTER - 1 WORD
1460			
1461	000110	DVVTEP= +72.	; VECTOR TABLE EXTEN. REL POINTER - 1 WORD
1462			
1463	000112	DVCTEP= +74.	; COMPILER TBL EXTEN. REL POINTER - 1 WORD
1464			
1465	000114	DVIWSP= +76.	; DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
1466			
1467	000116	DRTEND= +78.	; END OF DEVICE ROUTINE TABLE
1468			
1469			
1470	000001	.END	

ACTIVE= 100000		DEVDER= 000050		DVRGDT 004610R	002	LSPFWD 000742R	002	PTM4 = 000066	
ATIMSG 004505R	002	DEVORA= 000024		DVRGMG 004602R	002	LSPREV 000742R	002	PTM5 = 000070	
AURPEP= 000010		DEVETP= 000104		DVRINT= 000074		LSTATS 000732R	002	PTM6 = 000072	
AUTORP= 000020		DEVFWD= 000002		DVSFWD= 000064		LWAIT 000732R	002	PTM7 = 000074	
BCMRD 004632R	002	DEVI08= 000046		DVSVEC= 000066		LWEIRG 000733R	002	PTM8 = 000076	
BCMWR 004654R	002	DEVIVA= 000026		DVTVEC= 000072		LWREOF 000732R	002	PTM9 = 000100	
BINASC 000056R	002	DEVIW1= 000004		DVUPRT= 000052		MISCNT 001010R	002	PTEND = 000242	
BIT11 = 004000		DEVIW2= 000006		DVVTEP= 000110		MISCOM 002100R	002	PTLGTH= 000242	
BIT12 = 010000		DEVIW3= 000010		EOF 000010R	002	NOCOMP= 000001		PTCNT= 000030	
BIT6 = 000100		DEVIW4= 000012		EOFCNT 001016R	002	NOWAIT 001632R	002	PTSIZE= 000240	
BPI 001662R	002	DEVIW5= 000014		EOT 000012R	002	NUMRB 001054R	002	PUSRPC= 000236	
BPIER 001726R	002	DEVIW6= 000016		EOTCNT 001020R	002	OCPRES= 000100		PUTBYT 000100R	002
BPIVVL 001752R	002	DEVIW7= 000020		ERCOTB 003716R	002	ODD 001642R	002	PWRIOA= 000020	
BTASLZ 000060R	002	DEVIW8= 000022		ERMBAS 003302R	002	OFFLIN 002132R	002	RBADR 001050R	002
BYRD 000774R	002	DEVOK 002346R	002	ERR 000022R	002	PASCIN= 000006		RBCMD 001046R	002
BYWR 001000R	002	DEVVPS= 000030		ERRADR 001034R	002	PC =%000007		RBCNT 001052R	002
CIOSBY 000046R	002	DEVRSZ= 000000		ERRCNT 001022R	002	PCURDV= 000035		RBEXHM 005273R	002
CKDBSY 003076R	002	DEVSTP= 000102		ERRCOM 003222R	002	PDNUMS= 000036		RDCNT 001004R	002
CLIST 000054R	002	DEVWPS= 000032		ERRCS 003176R	002	PDPNTR= 000034		RDCOM 001774R	002
CLRVEC 000070R	002	DFLGWD 000002R	002	ERRCS1 003204R	002	PDST = 000122		RDRB 000004R	002
CMDCMS 004737R	002	DHKPAD= 000034		ERREX 003406R	002	PFBBOV= 000002		READ 001764R	002
CMDCOM 002166R	002	DISCNT 001314R	002	ERRIS 003214R	002	PFLGWD= 000000		RENDMG 004533R	002
CMDCRD 004707R	002	DISCT1 001320R	002	ERRSNM 003330R	002	PFWADR= 000004		REPORT 001134R	002
CMDCWR 004722R	002	DISPST 004236R	002	ERSTAD 003322R	002	PLNGTH= 000026		REPTBL 001430R	002
CMDEX 002460R	002	DISUIV 004204R	002	ERSTOP= 000004		PMDLCD= 000032		RESREG 004056R	002
CNTADR 001036R	002	DISUML 004232R	002	EVEN 001652R	002	PNAME = 000010		REWIND 002120R	002
CNTDER 005112R	002	DISUNM 004146R	002	FINCNT 001044R	002	PNBR = 000116		RINTEX 004010R	002
CNTEOF 005030R	002	DISUPR 004224R	002	GETBYT 000076R	002	PNMSG 004500R	002	RINTV 003766R	002
CNTEOT 005046R	002	DKILAD= 000040		GTNXTD= 001000		POBJST= 000024		RPTBAS 001364R	002
CNTERR 005075R	002	DOERCK= 000400		HSKEEP 001056R	002	POPSW = 000002		RPTEND 001410R	002
CNTINT 005140R	002	DOIOT = 000040		HSKPEN= 001056R	002	PRDIOA= 000016		RPTLP 001346R	002
CNTRRB 004770R	002	DREGAD 000024R	002	HSKPEP= 000004		PRINT 004364R	002	RRBCNT 001012R	002
CNTSEN= 005146R	002	DRTEND= 000116		HSKPST= 000744R	002	PROCTM 003410R	002	RTNINT 000074R	002
CNTSMG 004616R	002	DRTLTH= 000116		IFEOF 005171R	002	PROGNM 004502R	002	R0 =%000000	
CNTWRB 005003R	002	DTOEAD= 000044		IFEOT 005205R	002	PRONER= 020000		R1 =%000001	
CODFLD 005250R	002	DVBODA= 000060		INFOMG 005164R	002	PRTX 004476R	002	R2 =%000002	
CRESET 002144R	002	DVCMD5 000162R	002	INTCNT 001026R	002	PRTIWD 004326R	002	R3 =%000003	
CRLF 004670R	002	DVCprt= 000054		INVBPI 005322R	002	PSRC = 000120		R4 =%000004	
CSTAT 000760R	002	DVCPTe 000534R	002	INVDVN 005310R	002	PSRCST= 000022		R5 =%000005	
CSYSFW 000064R	002	DVCTEP= 000112		IOTO 005146R	002	PSTKCT= 000124		SAVREG 004042R	002
CTPRIO= 000020		DVCVEC= 000070		ISTAT 000744R	002	PSTKSV= 000126		SETDED= 000040	
CUPGER 000050R	002	DVGETB= 000076		IVCTAD 000026R	002	PSVREG= 000222		SETVEC 000066R	002
CURCNT 001042R	002	DVIWSP= 000114		KILL 001564R	002	PSWD 000030R	002	SIZE 000020R	002
CURFLG 001040R	002	DVIWST 000700R	002	KILLEX 001610R	002	PTEMO = 000056		SP =%000006	
CURMSG 004521R	002	DVMVTE 000444R	002	LBPI 000742R	002	PTM1 = 000060		SPCOM 002074R	002
CYCDVL= 002000		DVPDTA= 000062		LCOUNT 000732R	002	PTM10= 000102		SPFWD 002070R	002
CYCPRG= 040000		DVPKTE 000264R	002	LCRST 000732R	002	PTM11= 000104		SPOPER= 000200	
DATAER 001024R	002	DVPTEP= 000106		LEVEN 000732R	002	PTM12= 000106		SPREV 002112R	002
DECASC 000062R	002	DVPUTB= 000100		LNWAIT 000732R	002	PTM13= 000110		STMNG 005214R	002
DECTAD= 000042		DVREGE= 000162R	002	LODZ 000000R	002	PTM14= 000112		STMNUM 005224R	002
DERPAD= 000036		DVREGS 000116R	002	LODD 000732R	002	PTM15= 000114		STONER= 100000	
DError 002324R	002	DVREND= 005340R	002	LOFFLN 000732R	002	PTM2 = 000062		STSADR 003162R	002
DEVBTA= 000056		DVREX 001420R	002	LRWIND 000732R	002	PTM3 = 000064		STSTAT 004120R	002

SUPTAD	004074R	002	TOUTEX	001562R	002	UNITMG	004550R	002	WRCOM	002024R	002	WT410T=	000010	
TMEMSG	005232R	002	TSTVEC	000072R	002	URSTOP=	000002		WREIRG	002044R	002	XXXX	= 000000	
TMINT	002470R	002	TVECT	004012R	002	USEUBM=	000200		WREOF	002056R	002	.	= 005340R	002
. ABS.	000000	000												
	000000	001												
TM11	005340	002												

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.DTTMAB/NL:TOC/DOC=DTTMAB.P11
RUN-TIME: 4 9 1 SECONDS
RUN-TIME RATIO: 22/14=1.5
CORE USED: 5K (9 PAGES)

DOCUMENT PAGES: 32

