

# TC11

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

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

NOV 1976  
**digital**  
MADE IN U.S.A.











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  
110  
111

000000'

.SBTTL STANDARD DEVICE ROUTINE TABLE

.TITLE MAINDEC-11-DTTCA-B TC11/TU56 DEVICE ROUTINE FOR MPG

;REVISION 'B'

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

;MACY11: DTTCA? DTTCA?/CRF:SYM/DOC=DTTCA?.P11

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

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

.CSECT TC11  
.DSABL GBL

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

000000' 005276  
000002' 000000  
  
000004' 000000  
000006' 000000  
000010' 000000  
000012' 000000  
000014' 000000  
000016' 000000  
000020' 000001  
000022' 000000  
000024' 177340  
000026' 000214  
000030' 000300  
000032' 000000  
000034' 000710  
000036' 000742  
000040' 001360  
000042' 000654  
000044' 001254  
000046' 000000  
000050' 000000  
000052' 000000  
000054' 000000  
000056' 000000  
000060' 000000  
000062' 000000  
000064' 000000  
000066' 000000  
000070' 000000  
000072' 000000

LOCZ: .WORD DVREND-.  
DFLGWD: .WORD 0  
  
BLK: .WORD 0  
.WORD 0  
.WORD 0  
.WORD 0  
.WORD 0  
.WORD 0  
SIZE: .WORD 1  
ERR: .WORD 0  
DREGAD: .WORD 177340  
IVCTAD: .WORD 214  
PSWD: .WORD 300  
.WORD 0  
HSKEEP-.  
REPORT-.  
KILL-.  
DATAER-.  
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  
CLRVEC: .WORD 0  
TSTVEC: .WORD 0

:DEVICE ROUT SIZE IN BYTES  
:DEVICE ROUT FLAGWORD  
: BIT 15 = "NOWAIT" FLAG  
: BIT 11 = 0 - FWD, 1 - REV  
: BIT 3 = BLK SRCH ERROR  
: BIT 1 = DO I/O TERMINATION  
: BIT 0 = ERROR ON I/O CMND  
:CURRENT STARTING BLOCK NUMBER  
: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 6)  
: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  
:CLEAR INT VECTOR ROUT BR ADR  
:TEST INT VECTOR ROUT BR ADR







168	000264'	375	000		.BYTE	375,0	
169	000266'	052123	052101	051525	.ASCII	/STATUS/	
170	000274'	374	000		.BYTE	374,0	
171	000276'	047503	047125	051524	.ASCII	/COUNTS/	
172	000304'	373	000		.BYTE	373,0	
173	000306'	020040	043040	042127	.ASCII	/ FWD/	
174	000314'	372	000		.BYTE	372,0	
175	000316'	020040	051040	053105	.ASCII	/ REV/	
176	000324'	371	000		.BYTE	371,0	
177	000326'	051040	047104	046525	.ASCII	/ RDNUM/	
178	000334'	370	000		.BYTE	370,0	
179	000336'	051040	040504	046114	.ASCII	/ RDALL/	
180	000344'	367	000		.BYTE	367,0	
181	000346'	053440	040522	046114	.ASCII	/ WRALL/	
182	000354'	366	000		.BYTE	366,0	
183	000356'	020040	051127	046524	.ASCII	/ WRTM/	
184	000364'	365	000		.BYTE	365,0	
185	000366'	020040	052123	050117	.ASCII	/ STOP/	
186	000374'	364	000		.BYTE	364,0	
187	000376'	052123	040520	046114	.ASCII	/STPALL/	
188	000404'	363	000		.BYTE	363,0	
189							
190	000406'	000376	000632		DVMVTE: .WORD	376,LNWAIT-LOCZ	;MODEL VECTOR TABLE EXTEN.
191	000412'	000375	000632		.WORD	375,LWAIT-LOCZ	
192	000416'	000374	000632		.WORD	374,LSTATS-LOCZ	
193	000422'	000373	000632		.WORD	373,LCOUNT-LOCZ	
194	000426'	000372	000632		.WORD	372,LFWD-LOCZ	
195	000432'	000371	000632		.WORD	371,LREV-LOCZ	
196	000436'	000370	000633		.WORD	370,LRDNUM-LOCZ	
197	000442'	000367	000633		.WORD	367,LRDALL-LOCZ	
198	000446'	000366	000642		.WORD	366,LWRALL-LOCZ	
199	000452'	000365	000642		.WORD	365,LWRTM-LOCZ	
200	000456'	000364	000632		.WORD	364,LSTOP-LOCZ	
201	000462'	000363	000632		.WORD	363,LSTALL-LOCZ	
202							
203							
204							
205							
206	000466'	003	376		DVCPTE: .BYTE	3,376	;NO WAIT
207	000470'	004537	000012		.WORD	4537,10.	
208	000474'	003	375		.BYTE	3,375	;WAIT
209	000476'	004537	000012		.WORD	4537,10.	
210	000502'	004	374		.BYTE	4,374	;STATUS
211	000504'	004537	000012	001002	.WORD	4537,10.,1002	
212	000512'	004	373		.BYTE	4,373	;COUNTS
213	000514'	004537	000012	001001	.WORD	4537,10.,1001	
214	000522'	003	372		.BYTE	3,372	;FORWARD
215	000524'	004537	000012		.WORD	4537,10.	
216	000530'	003	371		.BYTE	3,371	;REVERSE
217	000532'	004537	000012		.WORD	4537,10.	
218	000536'	005	370		.BYTE	5,370	;READ NUMBER
219	000540'	004537	000012	000000	.WORD	4537,10.,0,2	
	000546'	000002					
220	000550'	005	367		.BYTE	5,367	;READ ALL
221	000552'	004537	000012	000000	.WORD	4537,10.,0,2	
	000560'	000002					



222	000562'	005	366	000000	.BYTE	5,366		;WRITE ALL
223	000564'	004537	000012	000000	.WORD	4537,10.,0,2		
	000572'	000002						
224	000574'	005	365	000000	.BYTE	5,365		;WRITE TIMING & MARK
225	000576'	004537	000012	000000	.WORD	4537,10.,0,2		
	000604'	000002						
226	000606'	003	364		.BYTE	3,364		;STOP
227	000610'	004537	000012		.WORD	4537,10.		
228	000614'	003	363		.BYTE	3,363		;STOP ALL
229	000616'	004537	000012		.WORD	4537,10.		
230								
231								
232								
233								
234	000622'	046102	020113		DVIWST: .ASCII	/BLK /		
235	000626'	000004			.WORD	DEVIW1		
236	000630'	177777			.WORD	177777		;END OF TABLE
237								
238								
239								
240								
241	000632'				LNWAIT:			
242	000632'				LWAIT:			
243	000632'				LSTATS:			
244	000632'				LCOUNT:			
245	000632'				LFWD:			
246	000632'				LREV:			
247	000632'				LSTOP:			
248	000632'	000			LSTALL: .BYTE	0		
249	000633'				LRDNUM:			
250	000633'	377	047111	047524	LRDALL: .ASCIZ	<377>/INT0/<377>		
	000640'	000377						
251	000642'				LWRALL:			
252	000642'	043377	047522	177515	LWRM: .ASCIZ	<377>/FROM/<377>		
	000650'	000						
253		000652'						
254								
255		000652'			HSKPST=	.		
256		000652'			ISTAT=	.		;STORAGE FOR DEV REG'S AT INT
257	000652'	000000			.WORD	0		
258	000654'	000000			.WORD	0		
259	000656'	000000			.WORD	0		
260	000660'	000000			.WORD	0		
261	000662'	000000			.WORD	0		
262								
263	000664'	000005			CSTAT: .BLKW	5		;DEV REG CURRENT VALUES STORAGE
264								
265	000676'	000000			BYRD: .WORD	0		;BYTES READ COUNT (READ, RDALL, & RDNUM)
266	000700'	000000			.WORD	0		
267	000702'	000000			BYWR: .WORD	0		;BYTES WRITTEN COUNT (WRITE, WRALL, &
268	000704'	000000			.WORD	0		WRTM)
269	000706'	000000			RDCNT: .WORD	0		;READ CMND COUNT (READ, RDALL, & RDNUM)
270	000710'	000000			WRCNT: .WORD	0		;WRITE CMND COUNT (WRITE, WRALL, & WRTM)
271	000712'	000000			MISCNT: .WORD	0		;MISC. CMND COUNT (STOP & STPALL)
272	000714'	000000			ERRCNT: .WORD	0		;DEVICE ERRORS COUNT
273	000716'	000000			DATAER: .WORD	0		;DATA ERRORS COUNT



274	000720'	000000	INTCNT: .WORD	0	;INTERRUPTS COUNT
275					
276	000722'	000000	TOECNT: .WORD	0	;# OF ENTRIES INTO T/O ERROR ROUT
277	000724'	000000	ERRADR: .WORD	0	;CURR ADR IN USER PROG
278	000726'	000000	CNTADR: .WORD	0	;ADR OF BYTE COUNT TOTALS
279	000730'	000000	CURFLG: .WORD	0	;FLAG WORD OF CURR CMND
280	000732'	000000	CURCNT: .WORD	0	;WORD CNT FOR CURR CMND
281	000734'	000000	CURCMD: .WORD	0	;CURRENT BLK ORIENTED CMND
282	000736'	000000	INBLKN: .WORD	0	;INITIAL BLK # FOR SEARCH
283	000740'	000000	FINCNT: .WORD	0	;FINAL WORD COUNT (TCWC)
284	000742'	000000	REVCNT: .WORD	0	;TAPE DIRECTION REVERSAL CNT
285		000744'	HSKPEN= .		
286					
287		000000	XXXX=	0	;VALUE TO BE TAILORED BY DEV ROUT



289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344

.SBTTL TC11 SUPPORT ROUTINES ENTERED FROM MPG

;DEVICE ROUTINE HOUSEKEEPING

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

HSKEEP: TST (R5)+ ;UNCONDITIONALLY DO HSKP?  
BNE 10\$ ;N,Y-10\$  
BIT #HSKPST,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

;TC11 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 #10,R1 ;GET # OF CNT WORDS  
5\$: TST (R0)+ ;THIS CNT WORD = 0?  
BNE 10\$ ;Y,N-10\$  
DEC R1 ;DECR WORD CNT  
BNE 5\$ ;CK'ED ALL WORDS? (Y,N-5\$)  
BR DVREX ;GO TO EXIT -- ALL CNTS ARE 0'S  
10\$: JSR PC,SUPTAD ;SET UP PROG TBL ADR IN R3  
MOV (R5)+,R4 ;GET FLAGWORD  
BIT #2,R4 ;GOING TO DO STATUS DISPLAY?  
BEQ DISCNT ;Y,N-DISCNT  
337 JSR R5,STSTAT ;GO STORE STATUS REG'S  
338 .WORD CSTAT-  
339 MOV PC,R0 ;SET UP ADR OF REG'S AT  
340 ADD #ISTAT-,R0 ;LAST INT  
341 MOV #5,R1 ;SET UP # OF REG'S  
20\$: TST (R0)+ ;ALL REG'S = 0?  
BNE 30\$ ;N,Y-40\$  
DEC R1



345	001100'	001374		BNE	20\$	
346	001102'	000412		BR	40\$	
347	001104'	004767	003116	30\$: JSR	PC,DISUNM	;DISPLAY CURR UNIT #
348	001110'	004567	003316	.JSR	R5,PRINT	;ISSUE 'AT LAST INT' MSG
349	001114'	003437		.WORD	ATIMSG-	
350	001116'	000031		.WORD	25.	
351	001120'	004567	003172	.JSR	R5,DISPST	;GO DISPLAY STATUS AT LAST INT
352	001124'	177526		.WORD	ISTAT-	
353	001126'	000402		BR	45\$	;CONTINUE DISPLAY
354	001130'	004767	003072	40\$: JSR	PC,DISUNM	;DISPLAY CURR UNIT #
355	001134'	004567	003272	45\$: JSR	R5,PRINT	;ISSUE 'CURRENTLY' MSG
356	001140'	003444		.WORD	CURMSG-	
357	001142'	000012		.WORD	10.	
358	001144'	004567	003146	.JSR	R5,DISPST	;GO DISPLAY CURRENT STATUS
359	001150'	177514		.WORD	CSTAT-	
360	001152'	004767	003230	.JSR	PC,PRTIWD	;GO DISPLAY INFO WORDS
361	001156'	000402		BR	DISCT1	;CHECK FOR COUNTS DISPLAY
362	001160'	004767	003042	DISCNT: JSR	PC,DISUNM	;DISPLAY CURR UNIT #
363	001164'	032704	000001	DISCT1: BIT	#1,R4	;DISPLAY COUNTS?
364	001170'	001431		BEQ	RPTEND	;Y,N-RPTEND
365	001172'	012700	000012	MOV	#10,R0	;SET UP # OF WORDS
366	001176'	010701		MOV	PC,R1	;SET UP ADR OF CNTS
367	001200'	062701	177476	ADD	#BYRD-.,R1	
368	001204'	010702		MOV	PC,R2	;SET UP TBL ADR
369	001206'	062702	000066	ADD	#REPTBL-.,R2	
370	001212'	012267	000012	RPTLP: MOV	(R2)+,RPTBAS	;MOV MSG ADR TO S/R LINKAGE
371	001216'	004067	002702	.JSR	R0,SAVREG	;SAVE ALL REG'S
372	001222'	011100		MOV	(R1),R0	;GET CURRENT COUNT
373	001224'	004577	176626	RPTBAS: JSR	R5,ABINASC	;CONVERT IT TO ASCII
374	001230'	000000		.WORD	XXXX	
375	001232'	004067	002702	.JSR	R0,RESREG	;RESTORE REG'S
376	001236'	005721		TST	(R1)+	;POINT AT NXT CNT
377	001240'	005300		DEC	R0	;DONE ALL WORDS?
378	001242'	001363		BNE	RPTLP	;Y,N-RPTLP
379	001244'	004567	003162	.JSR	R5,PRINT	;GO ISSUE COUNTS MSG
380	001250'	003434		.WORD	CNTSMG-	
381	001252'	000221		.WORD	CNTSEN-CNTSMG	
382	001254'	004567	003152	RPTEND: JSR	R5,PRINT	;ISSUE "END OF REPORT" MSG
383	001260'	003336		.WORD	RENDMG-	
384	001262'	177763		.WORD	-13.	
385	001264'	004067	002650	DVREX: JSR	R0,RESREG	;RESTORE REGISTERS
386	001270'	005725		TST	(R5)+	;SET UP RETURN POINT
387	001272'	000205		RTS	R5	;EXIT IN-LINE
388						
389						
390	001274'	003470		REPTBL: .WORD	BCMRD-RPTBAS	
391	001276'	003476		.WORD	BCMRD+6-RPTBAS	
392	001300'	003512		.WORD	BCMWR-RPTBAS	
393	001302'	003520		.WORD	BCMWR+6-RPTBAS	
394	001304'	003545		.WORD	CMDCRD-RPTBAS	
395	001306'	003560		.WORD	CMDCWR-RPTBAS	
396	001310'	003575		.WORD	CMDCMS-RPTBAS	
397	001312'	003624		.WORD	CNTERR-RPTBAS	
398	001314'	003641		.WORD	CNTDER-RPTBAS	
399	001316'	003667		.WORD	CNTINT-RPTBAS	
400						



```

401
402
403
404
405
406 001320' 005267 177376
407 001324' 026727 177372 000007
408 001332' 001031
409 001334' 004067 002564
410 001340' 004767 002612
411 001344' 004567 002632
412 001350' 177314
413 001352' 004567 002516
414 001356' 000404
415 001360' 112714 000011
416 001364' 004767 002460
417 001370' 042713 000010
418 001374' 004567 001674
419 001400' 001553
420 001402' 000016
421 001404' 004067 002530
422 001410' 012605
423 001412' 000177 176432
424 001416' 000205
425
426
427
428
429
430
431
432
433 001420' 004567 002450
434 001424' 000407
435 001426' 016701 176372
436 001432' 112761 000011 000002
437 001440' 004767 002404
438 001444' 000205

;TIMEOUT ERROR ROUTINE
;JSR R5,TOUTER S/R CALL
TOUTER: INC TOECNT ;INCR # OF TIMEOUTS THAT OCCURRED
CMP TOECNT,#7 ;THIS SEVENTH TIMEOUT IN A ROW?
BNE TOUTEX ;Y,N-TOUTEX
JSR RO,SAVREG ;SAVE ALL REGISTERS
JSR PC,SUPTAD ;SET UP TCCM & PROG TBL ADR'S
JSR R5,STSTAT ;STORE CURRENT STATUS
.WORD CSTAT-
JSR R5,TVECT ;CK IF I HAVE VECTOR CONTROL
BR 10$ ;BR IF I DON'T
MOVB #11,(R4) ;RESET INT ENABLE & STOP THE TAPE
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 IOTO-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 & STOP THE TAPE
JSR PC,RINTV ;RESET INT VECTOR INFO
KILLEX: RTS R5 ;EXIT IN-LINE

```



```

440 .SBTTL TC11 FUNCTION ROUTINES
441
442
443 ;"WAIT" FUNCTION ROUTINE
444
445 ;JSR R5, WAIT FUNCTION CALL
446
447 001446' 042767 100000 176326 WAIT: BIC #100000, DFLGWD ;RESET THE "NOWAIT" FLAG
448 001454' 004767 001506 JSR PC, CKDBSY ;WAIT IF BUSY & DO TERMINATION
449 001460' 004767 002364 JSR PC, RINTV ;RESET THE INTERRUPT VECTOR
450 001464' 000205 RTS R5 ;EXIT IN-LINE
451
452
453 ;"NOWAIT" FUNCTION ROUTINE
454
455 ;JSR R5, NOWAIT FUNCTION CALL
456
457 001466' 052767 100000 176306 NOWAIT: BIS #100000, DFLGWD ;SET THE "NOWAIT" FLAG
458 001474' 000205 FUNCEX: RTS R5 ;EXIT IN-LINE
459
460
461 ;"FWD" FUNCTION ROUTINE
462
463 ;JSR R5, FWD FUNCTION CALL
464
465 001476' 042767 004000 176276 FWD: BIC #4000, DFLGWD ;RESET THE REVERSE FLAG
466 001504' 000205 RTS R5 ;EXIT IN-LINE
467
468
469 ;"REV" FUNCTION ROUTINE
470
471 ;JSR R5, REV FUNCTION CALL
472
473 001506' 052767 004000 176266 REV: BIS #4000, DFLGWD ;SET THE REVERSE FLAG
474 001514' 000205 RTS R5 ;EXIT IN-LINE
475
476
477 ;"READ" FUNCTION ROUTINE
478
479 ;JSR R5, READ FUNCTION CALL
480 ;.WORD ADR DATA ADDRESS (BITS 16 & 17)
481 ;.WORD ADR DATA ADDRESS (BITS 0 - 15)
482 ;.WORD CNT BYTE COUNT
483 ;.WORD DEV (NOT USED)
484
485 001516' 012701 000105 READ: MOV #105, R1 ;SET UP READ CMND CODE
486 001522' 012702 000011 MOV #011, R2 ;SET UP READ FLAG WORD
487 001526' 004767 001434 RDCOM: JSR PC, CKDBSY ;GO CK IF DEV IS BUSY
488 001532' 005267 177150 INC RDCNT ;ADD 1 TO READ CMND CNT
489 001536' 010700 MOV PC, R0 ;SET UP ADR OF BYTES READ CNT
490 001540' 062700 177140 ADD #BYRD+2-., R0
491 001544' 000456 BR CMDCOM ;GO TO CMND COMMON PROCESSING
    
```



# MO1

```

493                                     ;"WRITE" FUNCTION ROUTINE
494
495                                     ;JSR      R5,WRITE      FUNCTION CALL
496                                     ;.WORD   ADR          DATA ADDRESS (BITS 16 & 17)
497                                     ;.WORD   ADR          DATA ADDRESS (BITS 0 - 15)
498                                     ;.WORD   CNT          BYTE COUNT
499                                     ;.WORD   DEV          (NOT USED)
500
501 001546' 012701 000115      WRITE:  MOV      #115,R1      ;SET UP WRITE CMND CODE
502 001552' 012702 000011      MOV      #011,R2      ;SET UP CMND FLAG WORD
503 001556' 004767 001404      WRCOM:  JSR      PC,CKOBSY ;GO CK IF DEV IS BUSY
504 001562' 005267 177122      INC      WRCNT        ;ADD 1 TO WRITE CMND CNT
505 001566' 010700              MOV      PC,R0         ;SET UP ADR OF BYTES WRITTEN CNT
506 001570' 062700 177114      ADD      #BYWR+2--,R0
507 001574' 000442              BR       CMDCOM       ;GO TO CMND COMMON PROCESSING
508
509                                     ;"RDNUM" FUNCTION ROUTINE
510
511                                     ;JSR      R5,RDNUM     FUNCTION CALL
512                                     ;.WORD   ADR          DATA ADDRESS
513                                     ;.WORD   CNT          BYTE COUNT
514
515
516 001576' 012701 000103      RDNUM:  MOV      #103,R1 ;SET UP RDNUM CMND CODE
517 001602' 012702 000102      MOV      #102,R2      ;SET UP CMND FLAG WORD
518 001606' 000747              BR       RDCOM        ;GO TO COMMON READ PROCESSING
519
520                                     ;"RDALL" FUNCTION ROUTINE
521
522                                     ;JSR      R5,RDALL     FUNCTION CALL
523                                     ;.WORD   ADR          DATA ADDRESS
524                                     ;.WORD   CNT          BYTE COUNT
525
526
527 001610' 012701 000007      RDALL:  MOV      #007,R1 ;SET UP RDALL CMND CODE
528 001614' 012702 000232      MOV      #232,R2      ;SET UP CMND FLAG WORD
529 001620' 000742              BR       RDCOM        ;GO TO COMMON READ PROCESSING
530
531                                     ;"WRALL" FUNCTION ROUTINE
532
533                                     ;JSR      R5,WRALL     FUNCTION CALL
534                                     ;.WORD   ADR          DATA ADDRESS
535                                     ;.WORD   CNT          BYTE COUNT
536
537
538 001622' 012701 000017      WRALL:  MOV      #017,R1 ;SET UP WRALL CMND CODE
539 001626' 012702 000032      MOV      #032,R2      ;SET UP CMND FLAG WORD
540 001632' 000751              BR       WRCOM        ;GO TO COMMON WRITE PROCESSING
    
```



```

542                                     ;"WRTM" FUNCTION ROUTINE
543
544                                     ;JSR    R5,WRTM      FUNCTION CALL
545                                     ;.WORD  ADR        DATA ADDRESS
546                                     ;.WORD  CNT        BYTE COUNT
547
548 001634' 012701 000013      WRTM:  MOV    #013,R1      ;SET UP WRTM CMND CODE
549 001640' 012702 000022      MOV    #022,R2      ;SET UP CMND FLAG WORD
550 001644' 000744              BR     WRCOM        ;GO TO COMMON WRITE PROGRESSING
551
552                                     ;"STOP" FUNCTION ROUTINE
553
554                                     ;JSR    R5,STOP    FUNCTION CALL
555
556
557 001646' 012701 000111      STOP:  MOV    #111,R1  ;SET UP STOP CMND CODE
558 001652' 012702 000044      MOV    #044,R2      ;SET UP CMND FLAG WORD
559 001656' 004767 001304      MISCOM: JSR   PC,CKDBSY ;GO CK IF DEV IS BUSY
560 001662' 005267 177024      INC    MISCNT       ;ADD 1 TO MISC. CMND CNT
561 001666' 000405              BR     CMDCOM       ;GO TO CMND COMMON PROCESSING
562
563                                     ;"STPALL" FUNCTION ROUTINE
564
565                                     ;JSR    R5,STPALL  FUNCTION CALL
566
567
568 001670' 012701 000101      STPALL: MOV   #101,R1  ;SET UP STPALL CMND CODE
569 001674' 012702 000044      MOV   #044,R2      ;SET UP CMND FLAG WORD
570 001700' 000766              BR   MISC0M       ;GO TO MISC. CMND COM PROCESSING
    
```



;COMMAND COMMON PROCESSING ROUTINE

;R4 = ADR OF TCCM 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 7 = "RDALL" CMND  
;BIT 6 = "RDNUM" CMND  
;BIT 5 = DON'T ISSUE STOP AT FINAL INT  
;BIT 4 = NON-INTERRUPT DATA SERVICE  
;BIT 3 = BLOCK SEARCH BEFORE CMND  
;BIT 2 = DO NOT INCREMENT BYTE COUNTS  
;BIT 1 = 2 ARGUMENT CMND  
;BIT 0 = 4 ARGUMENT CMND

572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627

001702' 010067 177020  
001706' 010267 177016  
001712' 005067 176104  
001716' 032702 000003  
001722' 001430  
001724' 062704 000004  
001730' 032702 000001  
001734' 001406  
001736' 012500  
001740' 006300  
001742' 006300  
001744' 006300  
001746' 006300  
001750' 050001  
001752' 012514  
001754' 012544  
001756' 006214  
001760' 011467 176746  
001764' 011467 176750  
001770' 005414  
001772' 005744  
001774' 032702 000001  
002000' 001401  
002002' 005725  
002004' 010167 176724  
002010' 116300 000035  
002014' 020027 000007  
002020' 101405  
002022' 004567 001240  
002026' 001701  
002030' 000012  
002032' 000427  
002034' 110064 000001  
002040' 016700 175736  
002044' 042700 173777  
002050' 050014

CMDCOM: MOV R0,CNTADR  
MOV R2,CURFLG  
CLR ERR  
BIT #3,R2  
BEQ 10\$  
ADD #4,R4  
BIT #1,R2  
BEQ 5\$  
MOV (R5)+,R0  
ASL R0  
ASL R0  
ASL R0  
BIS R0,R1  
5\$: MOV (R5)+,(R4)  
MOV (R5)+,-(R4)  
ASR (R4)  
MOV (R4),CURCNT  
MOV (R4),FINCNT  
NEG (R4)  
TST -(R4)  
BIT #1,R2  
BEQ 10\$  
TST (R5)+  
10\$: MOV R1,CURCMD  
MOVB PCURDV(R3),R0  
CMP R0,#7  
BLOS 20\$  
JSR R5,ERRCS  
.WORD INVDVN-ERMBAS  
.WORD 10  
BR 30\$  
20\$: MOVB R0,1(R4)  
MOV DFLGWD,R0  
BIC #173777,R0  
BIS R0,(R4)

;SAVE ADR OF BYTE COUNT  
;SAVE FLAGWD FOR TERMINATION  
;RESET THE ERROR INDICATOR  
;THIS CMND HAVE ARGUMENTS?  
;Y,N-10\$  
;POINT AT BUS ADR REG  
;4 ARGUMENT CMND?  
;Y,N-5\$  
;GET BITS 16 & 17 OF BUS ADR  
;ALIGN THEM TO CORRECT  
;BIT POSITIONS  
  
;SET THEM INTO CMND CODE WORD  
;GET BUS ADR BITS 0 - 15  
;GET BYTE COUNT  
;MAKE IT A WORD COUNT  
;SAVE WORD COUNT  
;INITIALIZE FINAL CNT TO SAME  
;MAKE IT NEGATIVE  
;REALIGN REG ADR TO TCCM  
;4 ARGUMENT CMND?  
;Y,N-10\$  
;BYPASS FOURTH ARGUMENT  
;SAVE CURR CMND CODE  
;GET CURR DEV #  
;INV DEV #?  
;Y,N-20\$  
;GO REPORT INV DEV # ERROR  
  
;GO TO ERR RETN  
;PUT DEV # IN TCCM BITS 9 THRU 10  
;GET DEV ROUT FLGWD  
;RESET ALL BITS EXCEPT REV FLAG  
;SET UP TAPE DIRECTION



628	002052'	032702	000010		BIT	#10,R2	: THIS A BLK SEARCH TYPE OF CMND?	
629	002056'	001426			BEG	50\$	: Y,N-50\$	
630	002060'	016767	175720	176650	MOV	BLK,INBLKN	: INITIALIZE BLK # FOR SEARCH S/R	
631	002066'	012701	000103		MOV	#103,R1	: SET UP "RDNUM" CMND CODE	
632	002072'	026727	175706	001101	CMP	BLK,#577.	: IS BLOCK # VALID?	
633	002100'	101415			BLOS	50\$	: N,Y-50\$	
634	002102'	004567	001160		JSR	R5,ERRCS	: REPORT INV BLK # ERROR	
635	002106'	001713			.WORD	INVBKN-ERMBAS		
636	002110'	000011			.WORD	9.		
637	002112'	005267	176600		INC	DATAER	: ADD 1 TO DATA ERR CNT	
638	002116'	012767	000001	175676	MOV	#1,ERR	: SET THE ERROR INDICATOR	
639	002124'	005367	176564		DEC	ERRCNT	: REMOVE THE 1 ADDED TO DEV ERR CNT	
640	002130'	000177	175714		JMP	3CUPGR	: GO TO MPG ERR RETN POINT	
641	002134'	042767	000011	175640	BIC	#11,DFLGWD	: RESET THE ERROR FLAGS	
642	002142'	005067	176574		CLR	REVCNT	: RESET TAPE REVERSAL CNT	
643	002146'	032702	000020		BIT	#20,R2	: NON-INTERRUPT SERVICING?	
644	002152'	001025			BNE	NONINT	: N,Y-NONINT	
645	002154'	005063	000030		CLR	PTOCNT(R3)	: INITIALIZE TIMEOUT COUNTER	
646	002160'	005067	176536		CLR	TOECNT	: RESET # OF TIMEOUTS	
647	002164'	052767	000002	175610	BIS	#2,DFLGWD	: SET THE "PROCESS TERMINATION" FLAG	
648	002172'	052713	000010		BIS	#WT4IOT,(R3)	: SET WAITING FOR I/O TERM FLAG	
649	002176'	110114			MOVB	R1,(R4)	: ISSUE THE CMND	
650	002200'	005767	175576		TST	DFLGWD	: "NOWAIT" BIT SET?	
651	002204'	100405			BMI	WTNOT	: N,Y-WTNOT	
652	002206'	004577	175634		JSR	R5,3CIOBSY	: WAIT FOR I/O TO COMPLETE	
653	002212'	004767	001252		JSR	PC,PROCTM	: GO PROCESS TERMINATION	
654	002216'	000205			RTS	R5	: EXIT IN-LINE TO USER PROG	
655								
656	002220'	042713	000010		WTNOT:	BIC	#WT4IOT,(R3)	: RESET WAITING FOR I/O TERM
657	002224'	000774			BR	CMDEX	: GO TO EXIT	



```

        .SBTTL TC11 NON-INTERRUPT COMMAND & DATA SERVICING
659
660
661
662 002226' 042701 000100      NONINT: BIC      #100,R1      ;RESET INT ENABLE IN CMND
663 002232' 110114              MOVB     R1,(R4) ;ISSUE RDNUM/WRTM CMND
664 002234' 032702 000010      BIT      #10,R2  ;BLK SEARCH TYPE OF CMND?
665 002240' 001506              BEQ     70$     ;Y,N-70$
666 002242' 105714              10$: TSTB    (R4)  ;READY SET?
667 002244' 100376              BPL     10$     ;Y,N-10$
668 002246' 012701 000003      MOV     #3,R1   ;SET UP RDNUM CMND CODE
669 002252' 004767 000454      JSR     PC,SEARCH ;GO SEARCH FOR BLK
670 002256' 000430              BR      25$     ;ERROR ? (N,Y-25$)
671 002260' 032767 000010 176442 BIT     #10,CURFLG ;FOUND THE BLK?
672 002266' 001365              BNE     10$     ;Y,N-10$
673 002270' 004767 000222      JSR     PC,NINTSU ;GO SET UP REGISTERS
674 002274' 032702 000200      BIT     #200,R2 ;"RDALL" CMND?
675 002300' 001443              BEQ     50$     ;Y,N-50$
676
677                                ;"RDALL" SERVICING
678
679 002302' 105714              20$: TSTB    (R4)  ;READY SET?
680 002304' 100376              BPL     20$     ;Y,N-20$
681 002306' 005714              TST     (R4)    ;ERROR BIT SET?
682 002310' 100415              BMI     30$     ;N,Y-30$
683 002312' 016402 177776      MOV     -2(R4),R2 ;GET TCST REG WITH BITS 16 & 17
684 002316' 042702 177774      BIC     #177774,R2 ;RESET OTHER BITS
685 002322' 010220              MOV     R2,(R0)+ ;STORE WORD WITH BITS 16 & 17
686 002324' 005201              INC     R1      ;DECR NEG WORD CNT
687 002326' 001411              BEQ     40$     ;CNT = 0? (N,Y-40$)
688 002330' 011320              MOV     (R3),(R0)+ ;STORE BITS 0-15 FROM TCDT
689 002332' 005201              INC     R1      ;DECR NEG WORD CNT
690 002334' 001362              BNE     20$     ;CNT = 0? (Y,N-20$)
691 002336' 000405              BR      40$     ;GO TO TERMINATION
692
693                                ;NON-INT CMND TERMINATION
694
695 002340' 004767 000152      25$: JSR     PC,NINTSU ;SET UP CORRECT REG VALUES
696 002344' 052767 000001 175430 30$: BIS     #1,DFLGWD ;SET THE "ERROR" FLAG
697 002352' 010167 176362      40$: MOV     R1,FINCNT ;STORE FINAL WORD CNT
698 002356' 010164 000002      MOV     R1,2(R4) ;MOVE FINAL CNT TO TCWC
699 002362' 010064 000004      MOV     R0,4(R4) ;MOVE FINAL BUS ADR TO TCBA
700 002366' 004567 001610      JSR     R5,STSTAT ;STORE ALL STATUS REGISTERS
701 002372' 176260              .WORD   ISTAT-.
702 002374' 005713              TST     (R3)    ;KNOCK DOWN READY IF STILL UP
703 002376' 112714 000011      MOVB    #011,(R4) ;ISSUE "STOP" CMND
704 002402' 004767 001550      JSR     PC,SUPTAD ;RESTORE PROG TBL ADR
705 002406' 000701              BR      CMDEND  ;GO PROCESS TERMINATION & EXIT
706
707                                ;"WRALL" SERVICING
708
709 002410' 105714              50$: TSTB    (R4)  ;READY SET?
710 002412' 100376              BPL     50$     ;Y,N-50$
711 002414' 005714              TST     (R4)    ;ERROR BIT SET?
712 002416' 100752              BMI     30$     ;N,Y-30$
713 002420' 012002              MOV     (R0)+,R2 ;GET WORD WITH BITS 16 & 17
714 002422' 042702 177774      BIC     #177774,R2 ;RESET OTHER BITS

```



```

715 002426' 010264 177776      MOV      R2,-2(R4)      ;SET BITS 16 & 17 INTO TCST
716 002432' 005201              INC      R1            ;DECR NEG WORD CNT
717 002434' 001403              BEQ     60$           ;CNT = 0? (N,Y-60$)
718 002436' 012013              MOV     (R0)+,(R3)    ;MOVE BITS 0-15 WORD TO TCDT
719 002440' 005201              INC     R1            ;DECR NEG WORD CNT
720 002442' 001362              BNE    50$           ;CNT = 0? (Y,N-50$)
721 002444' 032754 001000 177776 60$: BIT     #1000,-2(R4)   ;DATA MISSED SET YET?
722 002452' 001774              BEQ     60$           ;Y,N-60$
723 002454' 000736              BR     40$           ;GO TO TERMINATION
724
725                                ;"WRTM" SERVICING
726
727 002456' 004767 000034      70$:   JSR     PC,NINTSU  ;SET UP REGISTERS
728 002462' 105714      80$:   TSTB   (R4)        ;READY SET YET?
729 002464' 100376              BFL    80$           ;Y,N-80$
730 002466' 005714              TST    (R4)        ;ERROR BIT SET?
731 002470' 100725              BMI    30$         ;N,Y-30$
732 002472' 005701              TST    R1          ;WORD CNT = 0?
733 002474' 001403              BEQ     90$         ;N,Y-90$
734 002476' 012013              MOV     (R0)+,(R3)  ;MOVE DATA WORD TO TCDT
735 002500' 005201              INC     R1          ;DECR NEG WORD CNT
736 002502' 000767              BR     80$         ;GO WAIT FOR NEXT WORD
737 002504' 012702 000310      90$:   MOV     #200.,R2   ;SET UP DELAY CNT
738 002510' 005302      100$:  DEC     R2          ;DELAY FEW HUNDRED MICROSEC'S
739 002512' 001376              BNE    100$        ;GO TO TERMINATION
740 002514' 000716              BR     40$
741
742                                ;NON-INT REGISTER SETUP S/R
743
744
745 002516' 016400 000004      NINTSU: MOV    4(R4),R0   ;MOVE BUS ADR TO R0
746 002522' 016401 000002      MOV    2(R4),R1   ;MOVE NEG WORD CNT TO R1
747 002526' 010403              MOV    R4,R3      ;SET UP TCDT ADR IN R3
748 002530' 062703 000006      ADD    #6,R3
749 002534' 000207              RTS     PC         ;EXIT IN-LINE

```



```

751                                     .SBTTL TC11 INTERRUPT SERVICE ROUTINE
752
753
754 002536' 004067 001362          TCINT: JSR      RD, SAVREG          ;SAVE ALL REGISTERS
755 002542' 005267 176152          INC      INT CNT          ;ADD 1 TO INTERRUPT CNT
756 002546' 004767 001404          JSR      PC, SUPTAD        ;SET UP PROG TBL & TCCM ADR'S
757 002552' 004567 001424          JSR      RS, STSTAT        ;STORE ALL DEV REG'S
758 002556' 176074                  .WORD    ISTAT-
759 002560' 016702 176144          MOV      CURFLG, R2        ;GET THIS CMND'S FLGWD
760 002564' 032702 000010          BIT      #10, R2          ;IN BLOCK SEARCH MODE?
761 002570' 001406                  BEQ      5$                ;Y, N-5$
762 002572' 012701 000103          MOV      #103, R1         ;SET UP RDNUM WITH INT ENABLE
763 002576' 004767 000130          JSR      PC, SEARCH        ;GO SEARCH FOR SPECIFIED BLK
764 002602' 000403                  BR       10$              ;ERROR ON SEARCH? (N, Y-10$)
765 002604' 000446                  BR       60$              ;GO TO INT EXIT
766 002606' 005714          5$: TST      (R4)            ;IS THERE AN ERR CONDITION?
767 002610' 100006                  BPL      30$              ;Y, N-30$
768 002612' 052767 000001 175162 10$: BIS      #1, DFLGWD        ;SET THE ERROR FLAG
769 002620' 112714 000011 20$: MOVB     #011, (R4)      ;STOP TAPE & RESET INT ENABLE
770 002624' 000431                  BR       50$              ;GO TO CMND TERMINATION
771
772                                     ;"RDNUM" USER CMND INT
773
774 002626' 032702 000100          30$: BIT      #100, R2      ;DOING A USER "RDNUM" CMND?
775 002632' 001421                  BEQ      40$              ;Y, N-40$
776 002634' 016401 000006          MOV      6(R4), R1        ;GET BLK # READ
777 002640' 016400 000004          MOV      4(R4), R0        ;GET CURRENT DATA ADR
778 002644' 004777 175230          JSR      PC, @PUTBYT      ;HAVE MPG STORE 1 BYTE
779 002650' 000301                  SWAB     R1                ;SET UP FOR 2ND BYTE
780 002652' 004777 175222          JSR      PC, @PUTBYT      ;STORE IT TOO
781 002656' 010064 000004          MOV      R0, 4(R4)        ;STORE NEW DATA ADR
782 002662' 005264 000002          INC      2(R4)            ;DECR NEG WORD CNT
783 002666' 001754                  BEQ      20$              ;CNT = 0? (N, Y-20$)
784 002670' 112714 000103          MOVB     #103, (R4)       ;ISSUE "RDNUM" CMND AGAIN
785 002674' 000412                  BR       60$              ;GO TO INTERRUPT EXIT
786
787                                     ;FINAL INTERRUPT PROCESSING
788
789 002676' 032702 000040          40$: BIT      #40, R2      ;ISSUE "STOP" CMND AT FINAL INT?
790 002702' 001746                  BEQ      20$              ;N, Y-20$
791 002704' 042714 000100          BIC      #100, (R4)        ;RESET INT ENABLE
792 002710' 016467 000002 176022 50$: MOV      2(R4), FINCNT    ;STORE FINAL WORD COUNT
793 002716' 042713 000010          BIC      #WT4IOT, (R3)    ;RESET WAITING FOR I/O TERM
794 002722' 004067 001212          60$: JSR      RD, RESREG    ;RESTORE ALL REGISTERS
795 002726' 000177 175142          JMP      @RTNINT          ;EXIT FROM INTERRUPT
    
```



797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852

.SBTTL TC11 SEARCH FOR DECTAPE BLOCK S/R

;SEARCH FOR SPECIFIED BLOCK

;JSR PC\_SEARCH S/R CALL  
;BR LABEL EXECUTED IF AN ERROR

;R1 = RDNUM CMND WITH OR W/O INT ENABLE  
;R3 = PROG TBL ADR  
;R4 = TCCM ADR  
;DESTROYS R0,R1

002732'	116300	000035		SEARCH: MOVB	PCURDV(R3),R0	;GET CURR DEV #
002736'	000300			SWAB	R0	;ALIGN DEV # BITS
002740'	050001			BIS	R0,R1	;SET DEV # INTO CMND
002742'	016700	175036		MOV	BLK,R0	;GET DESIRED BLOCK #
002746'	005714			TST	(R4)	;ERROR BIT SET?
002750'	100475			BMI	110\$	;N,Y-110\$
002752'	032767	004000	175022	BIT	#4000,DFLGWD	;DOING I/O'S IN REVERSE?
002760'	001046			BNE	100\$	;N,Y-100\$

;SEARCH FOR BLK FOR FWD I/O

002762'	026467	000006	175746	CMP	6(R4),INBLKN	;AT THE BLK WE'RE LOOKING FOR?
002770'	001430			BEQ	90\$	;N,Y-90\$
002772'	002421			BLT	80\$	;PAST THE BLOCK? (Y,N-80\$)
002774'	162700	000002		60\$: SUB	#2,R0	;SUB 2 FROM ORG BLK #
003000'	052701	004000		BIS	#4000,R1	;SET REV BIT IN CMND
003004'	032714	004000		BIT	#4000,(R4)	;ALREADY GOING REV?
003010'	001012			BNE	80\$	;N,Y-80\$
003012'	005267	175724		70\$: INC	REVCNT	;INCR REVERSAL CNT
003016'	026727	175720	000006	CMP	REVCNT,#6	;DONE 6 DIRECTION REVERSALS?
003024'	103404			BLO	80\$	;Y,N-80\$
003026'	052767	000010	174746	BIS	#10,DFLGWD	;SET BLK SRCH ERR FLG
003034'	000405			BR	88\$	;GO TO ERROR EXIT
003036'	010067	175674		80\$: MOV	R0,INBLKN	;SAVE BLK # (ORG OR ADJUSTED)
003042'	010114			85\$: MOV	R1,(R4)	;ISSUE DECTAPE CMND
003044'	062716	000002		ADD	#2,(SP)	;SET UP NORMAL RET ADR
003050'	000207			88\$: RTS	PC	;EXIT IN-LINE
003052'	032714	004000		90\$: BIT	#4000,(R4)	;ARE WE GOING FWD?
003056'	001346			BNE	60\$	;Y,N-60\$
003060'	105001			95\$: CLRB	R1	;CLEAR CMND CODE FIELD
003062'	156701	175646		BISB	CURCMD,R1	;SET IN REQUESTED CMND CODE
003066'	042767	000010	175634	BIC	#10,CURFLG	;RESET BLK SEARCH MODE FLAG
003074'	000762			BR	85\$	;GO ISSUE CMND

;SEARCH FOR BLK FOR REV I/O

003076'	052701	004000		100\$: BIS	#4000,R1	;INITIALIZE TO REV DIRECTION
003102'	026467	000006	175626	CMP	6(R4),INBLKN	;THIS THE BLOCK WE WANT?
003110'	001411			BEQ	105\$	;N,Y-105\$
003112'	003351			BGT	80\$	;IN FRONT OF THE BLK? (Y,N-80\$)
003114'	042701	004000		102\$: BIC	#4000,R1	;SET DIR TO FWD
003120'	062700	000002		ADD	#2,R0	;ALLOW FOR TURN AROUND
003124'	032714	004000		BIT	#4000,(R4)	;ALREADY GOING FWD?



853	003130'	001742		BEQ	80\$		:N, Y-80\$
854	003132'	000727		BR	70\$		:GO CK REVERSAL CNT
855	003134'	032714	004000	105\$: BIT	#4000, (R4)		:ARE WE GOING REV?
856	003140'	001347		BNE	95\$		:N, Y-95\$
857	003142'	000764		BR	102\$		:GO STORE ADJUSTED BLK #
858							
859	003144'	005764	177776	110\$: TST	-2(R4)		:END ZONE?
860	003150'	100337		BPL	88\$		:Y, N-88\$
861	003152'	032714	004000	BIT	#4000, (R4)		:GOING REV?
862	003156'	001315		BNE	70\$		:N, Y-70\$
863	003160'	052701	004000	BIS	#4000, R1		:SET DIRECTION TO REV
864	003164'	000712		BR	70\$		:GO CK REVERSAL CNT



```

866 .SBTTL SUBROUTINES FOR TC11 FUNCTION ROUTINES
867
868
869 ;CHECK IF DEVICE IS BUSY AND WAIT IF IT IS
870
871 ;JSR PC,CKDBSY S/R CALL
872
873 ;DESTROYS R0,R3,R4
874 ;ON EXIT:
875 ;R3 = PROG TBL ADR
876 ;R4 = TCCM ADR
877
878 003166' 004767 000764 CKDBSY: JSR PC,SUPTAD ;SET UP PROG TBL & TCCM ADR'S
879 003172' 032714 000100 10$: BIT #100,(R4) ;INT ENABLE ON?
880 003176' 001403 BEQ 20$ ;Y,N-20$
881 003200' 004577 174642 JSR R5,20IOBSY ;RELEASE CONTROL
882 003204' 000772 BR 10$ ;GO CK AGAIN
883 003206' 032767 000002 174566 20$: BIT #2,DFLGWD ;HAVE TO PROCESS PREV TERMINATION?
884 003214' 001403 BEQ 30$ ;Y,N-30$
885 003216' 004767 000246 JSR PC,PROCTM ;GO PROCESS TERMINATION
886 003222' 000763 BR 10$ ;GO RECHECK INT ENABLE
887 003224' 016767 174576 000012 30$: MOV IVCTAD,40$ ;STORE INT VECTOR ADR
888 003232' 016767 174572 000006 MOV PSWD,45$ ;STORE PROC STATUS WORD
889 003240' 004577 174622 JSR R5,2SETVEC ;GO SET UP THE VECTOR
890 003244' 000000 40$: .WORD XXXX ;INT VECTOR ADR
891 003246' 000000 45$: .WORD XXXX ;PSW
892 003250' 177266 .WORD TCINT- ;REL INT ROUT ADR
893 003252' 010567 175446 MOV R5,ERRADR ;SAVE CURR USER STMT ADR
894 003256' 162767 000004 175440 SUB #4,ERRADR
895 003264' 000207 RTS PC ;EXIT IN-LINE
896
897
898 ;ERROR INFORMATION DISPLAY S/R
899
900 ;JSR R5,ERRCS S/R CALL FOR CURR STATUS
901 ;JSR R5,ERRIS S/R CALL FOR INT STATUS
902 ;.WORD MSGADR-ERMBAS REL ADR OF ERROR MSG
903 ;.WORD MSGCNT # OF BYTES IN ERROR MSG
904 ;DESTROYS R0,R1,R2
905
906 003266' 004567 000710 ERRCS: JSR R5,STSTAT ;STORE CURR STATUS
907 003272' 175372 .WORD CSTAT-
908 003274' 012767 175262 000100 ERRCS1: MOV #CSTAT-ERSTAD,ERSTAD ;STORE ADR OF CURR STATUS
909 003302' 000403 BR ERRCOM ;GO TO COMMON POINT
910 003304' 012767 175250 000070 ERRIS: MOV #ISTAT-ERSTAD,ERSTAD ;STORE ADR OF LAST INT STATUS
911 003312' 012567 000034 ERRCOM: MOV (R5)+,ERMBAS ;STORE MSG ADR
912 003316' 012567 000032 MOV (R5)+,ERMBAS+2 ;STORE MSG CNT
913 003322' 005267 175366 INC ERRCNT ;ADD 1 TO ERROR CNT
914 003326' 032763 020000 000002 BIT #PRONER,POPSW(R3) ;ERROR PRINTING INHIBITED?
915 003334' 001054 BNE ERREX ;N,Y-ERREX
916 003336' 010446 MOV R4,-(SP) ;SAVE R4
917 003340' 005004 CLR R4 ;SET USER MODE PRINT FLAG
918 003342' 004767 000660 JSR PC,DISUNM ;DISPLAY UNIT #
919 003346' 004567 001060 JSR R5,PRINT ;PRINT ERROR MSG SPECIFIED
920 003352' 000000 ERMBAS: .WORD XXXX
921 003354' 000000 .WORD XXXX

```



```

922 003356' 026727 177770 001701      CMP      ERMBAS,#INVDVN-ERMBAS      ;INVALID UNIT # ERROR?
923 003364' 001411                      BEQ      ERRSNM                    ;N,Y-ERRSNM
924 003366' 026727 177760 001713      CMP      ERMBAS,#INVBKN-ERMBAS    ;INVALID BLOCK # ERROR?
925 003374' 001403                      BEQ      ERPBKN                    ;N,Y-ERPBKN
926 003376' 004567 000714                      JSR      RS,DISPST                 ;DISPLAY STATUS REG'S
927 003402' 000000                      ERSTAD: .WORD XXXX
928 003404' 004767 000776      ERPBKN: JSR      PC,PRTIWD          ;DISPLAY CURR BLK #
929 003410' 016300 000022      ERRSNM: MOV      PSRCST(R3),RO      ;GET ADR OF SRC STMENTS
930 003414' 111001                      10$:   MOVB     (RO),R1            ;SAVE STMT LENGTH
931 003416' 026067 000004 175300      CMP      4(RO),ERRADR            ;ERROR OCCUR ON THIS STMT?
932 003424' 001402                      BEQ      20$                      ;N,Y-20$
933 003426' 060100                      ADD     R1,RO                    ;POINT AT NXT STMT
934 003430' 000771                      BR      10$                      ;GO CK NXT STMT
935 003432' 005720                      20$:   TST     (RO)+              ;SET UP ADR OF STMT # DATA
936 003434' 010701                      MOV     PC,R1                    ;SET UP DATA OUTPUT ADR
937 003436' 062701 001546                      ADD     #STMNUM-. ,R1
938 003442' 004577 174414                      JSR     RS,2DECASC                ;CONVERT IT TO ASCII
939 003446' 012767 020040 001534      MOV     #20040,STMNUM+4          ;SET 2 LOW DIGITS TO SPACES
940 003454' 004567 000752                      JSR     RS,PRINT                  ;ISSUE STMT # MSG
941 003460' 001514                      .WORD  STMNMG-.
942 003462' 177762                      .WORD  -14.
943 003464' 012604                      MOV     (SP)+,R4                  ;RESTORE R4
944 003466' 000205      ERREX: RTS      R5                ;EXIT IN-LINE
945
946
947                                     ;PROCESS TERMINATION OF PREVIOUS I/O FUNCTION
948
949                                     ;JSR      PC,PROCTM          S/R CALL
950
951 003470' 004067 000430      PROCTM: JSR     RO,SAVREG           ;SAVE ALL REG'S
952 003474' 042767 000002 174300      BIC     #2,DFLGWD                ;RESET PROCESS TERMINATION FLAG
953 003502' 032767 000004 175220      BIT     #4,CURFLG                ;INCR BYTE COUNT?
954 003510' 001016                      BNE     6$                        ;Y,N-6$
955 003512' 016700 175214      MOV     CURCNT,RO                ;GET INITIAL WORD CNT
956 003516' 016701 175216      MOV     FINCNT,R1                ;GET FINAL WORD CNT
957 003522' 100001                      BPL     2$                        ;IS IT NEGATIVE? (Y,N-2$)
958 003524' 005401                      NEG     R1                        ;MAKE IT POSITIVE
959 003526' 160100                      2$:   SUB     R1,RO                ;SUB REMAINING CNT FROM INITIAL CNT
960 003530' 006300                      ASL     RO                        ;MAKE IT A BYTE CNT
961 003532' 010067 174262      MOV     RO,SIZE                  ;STORE # OF BYTES ACTUALLY XFERRED
962 003536' 016701 175164      MOV     CNTADR,R1                ;GET ADR OF BYTE CNT TOTALS
963 003542' 060011                      ADD     RO,(R1)                  ;ADD IN THIS CNT
964 003544' 005541                      ADC     -(R1)                    ;UPDATE MOST SIGNF WORD OF CNT
965 003546' 032767 000001 174226      6$:   BIT     #1,DFLGWD              ;WAS THERE AN ERROR?
966 003554' 001476                      BEQ     80$                       ;Y,N-80$
967 003556' 012767 000001 174236      MOV     #1,ERR                   ;SET THE ERROR INDICATOR
968 003564' 032763 000400 000002      BIT     #DOERCK,POPSW(R3)        ;SUPPOSED TO DO ERROR CHECKING?
969 003572' 001065                      BNE     70$                       ;Y,N-70$
970 003574' 032767 000010 174200      BIT     #10,DFLGWD               ;BLOCK SEARCH ERROR?
971 003602' 001070                      BNE     90$                       ;N,Y-90$
972 003604' 010701                      MOV     PC,R1                    ;GET ADR OF CODE AREA IN ERR MSG
973 003606' 062701 001422                      ADD     #CODFLD-. ,R1
974 003612' 010102                      MOV     R1,R2                    ;MOVE IT TO WORK REG
975 003614' 012700 000023                      MOV     #19,RO                   ;SET UP AREA SIZE
976 003620' 112722 000040      10$:   MOVB     #40,(R2)+              ;CLEAR AREA TO SPACES
977 003624' 005300                      DEC     RO
    
```







```

1030                                     ;RESET INTERRUPT VECTOR S/R
1031
1032                                     ;JSR   PC,RINTV      S/R CALL
1033                                     ;R3 MUST CONTAIN PROG TBL ADR
1034                                     ;DESTROYS R0
1035
1036 004050' 004567 000020      RINTV: JSR   R5,TVECT      ;GO CK IF I HAVE VECTOR CONTROL
1037 004054' 000406              BR    RINTEX      ;BR IF I DON'T
1038 004056' 016767 173744 000004  MOV   IVCTAD,10$ ;GET CURR INT VECT ADR
1039 004064' 004577 174000      JSR   R5,@CLAVEC ;GO HAVE MPG CLEAR IT
1040 004070' 000000      10$:  .WORD XXXX
1041 004072' 000207      RINTEX: RTS   PC      ;EXIT IN-LINE
1042
1043
1044                                     ;TEST INTERRUPT VECTOR S/R
1045
1046                                     ;JSR   R5,TVECT      S/R CALL
1047                                     ;BR    LABEL        EXECUTED IF NOT SAME
1048                                     ;R3 MUST CONTAIN PROG TBL ADR
1049                                     ;DESTROYS R0
1050
1051 004074' 016767 173726 000010  TVECT: MOV   IVCTAD,20$ ;GET CURR INT VECT ADR
1052 004102' 016346 000004              MOV   PFWADR(R3),-(SP) ;STORE FLGWD ADR TO IDENTIFY ME
1053 004106' 004577 173760      JSR   R5,@STVECT ;DO I HAVE VECTOR CONTROL?
1054 004112' 000000      20$:  .WORD XXXX ;MPG WILL TELL ME SINCE I CAN'T
1055 004114' 176422              .WORD TCINT- ;GET AT LOWER MEM IF MEM MGMNT
1056 004116' 000401              BR    TVECTX      ;BR IF I DONT'T HAVE CNTRL
1057 004120' 005725              TST   (R5)+ ;BYPASS BR INST IN S/R CALL
1058 004122' 000205      TVECTX: RTS   R5      ;EXIT IN-LINE

```



```

1060          .SBTTL  SUBROUTINES FOR TC11 DEVICE ROUTINE
1061
1062
1063
1064          ;SAVE REGISTERS R0 THRU R5
1065
1066          ;JSR    R0,SAVREG      S/R CALL
1067
1068 SAVREG: MOV    R1,-(SP)          ;SAVE R0 THRU R5
1069        MOV    R2,-(SP)
1070        MOV    R3,-(SP)
1071        MOV    R4,-(SP)
1072        MOV    R5,-(SP)
1073        MOV    R0,PC            ;EXIT IN-LINE
1074
1075
1076          ;RESTORE REGISTERS R0 THRU R5
1077
1078          ;JSR    R0,RESREG      S/R CALL
1079
1080 RESREG: TST    (SP)+            ;RESTORE R5 THRU R0
1081        MOV    (SP)+,R5
1082        MOV    (SP)+,R4
1083        MOV    (SP)+,R3
1084        MOV    (SP)+,R2
1085        MOV    (SP)+,R1
1086        RTS    R0              ;EXIT IN-LINE
1087
1088
1089          ;SET PROGRAM'S PROG TABLE ADR IN R3 & TCCM ADR IN R4
1090
1091          ;JSR    PC,SUPTAD      S/R CALL
1092
1093 SUPTAD: MOV    PC,R3            ;SET UP LOCATION ZERO ADR
1094        ADD    #LOCZ-,R3
1095        SUB    -2(R3),R3        ;SUBTRACT PROG TBL LENGTH
1096        MOV    DREGAD,R4       ;GET DEV REG BASE ADR
1097        ADD    #2,R4           ;POINT AT TCCM
1098        RTS    PC             ;EXIT IN-LINE
1099
1100
1101          ;STORE DEVICE'S STATUS REGISTERS
1102
1103          ;JSR    R5,STSTAT      S/R CALL
1104        ;.WORD  STADR-          REL STORAGE ADR
1105        ;DESTROYS R0,R1
1106
1107 STSTAT: MOV    R5,R1            ;GET REL STORAGE ADR & MAKE
1108        ADD    (R5)+,R1        ;IT ABSOLUTE
1109        MOV    DREGAD,R0       ;GET ADR OF DEV REG'S
1110        MOV    (R0)+,(R1)+    ;STORE ALL DEV REG'S
1111        MOV    (R0)+,(R1)+
1112        MOV    (R0)+,(R1)+
1113        MOV    (R0)+,(R1)+
1114        MOV    (R0),(R1)
1115        RTS    R5            ;EXIT IN-LINE

```



```

1116
1117
1118                                     ;DISPLAY CURRENT UNIT #
1119
1120                                     ;JSR   PC,DISUNM      S/R CALL
1121                                     ;R3 MUST CONTAIN PROG TBL ADR
1122                                     ;DESTROYS R0,R1,R2
1123
1124 004226' 012767 000031 000056 DISUNM: MOV      #25,DISUML      ;INITIALIZE TO NORMAL MSG LNGTH
1125 004234' 116300 000035          MOVB    PCURDV(R3),R0    ;GET CURR UNIT #
1126 004240' 020027 000007          CMP     R0,#7          ;VALID UNIT #?
1127 004244' 101007          BHI    DISUIV        ;Y,N-DISUIV
1128 004246' 004577 173606          JSR    R5,@BTASLZ    ;CONVERT # TO DECIMAL ASCII
1129 004252' 000410          .WORD  UNASCI-
1130 004254' 016767 000406 000400    MOV     UNASCI+4,UNASCI ;MOVE ASCII # TO 1ST TWO DIGITS
1131 004262' 000410          BR     DISUPR        ;GO ISSUE MSG
1132 004264' 012767 000035 000020 DISUIV: MOV     #29,DISUML      ;SET UP ERR COND MSG LNGTH
1133 004272' 042700 177400          BIC    #177400,R0    ;RESET HIGH BYTE
1134 004276' 004577 173554          JSR    R5,@BINASC   ;CONVERT BINARY # TO ASCII
1135 004302' 000360          .WORD  UNASCI-
1136 004304' 004567 000122          DISUPR: JSR    R5,PRINT ;GO ISSUE UNIT # MSG
1137 004310' 000323          .WORD  UNITMG-
1138 004312' 000031          DISUML: .WORD  25
1139 004314' 000207          RTS     PC          ;EXIT IN-LINE
1140
1141
1142                                     ;TAILOR STATUS MSG & PRINT IT
1143
1144                                     ;JSR   R5,DISPST      S/R CALL
1145                                     ;WORD  STATADR-      REL ADR OF STATUS DATA
1146                                     ;DESTROYS R0,R1,R2
1147
1148 004316' 010502          DISPST: MOV     R5,R2      ;GET REL DATA ADR
1149 004320' 062502          ADD     (R5)+,R2      ;MAKE IT ADR
1150 004322' 010701          MOV     PC,R1        ;SET UP ACK OF REG NAMES IN ASCII
1151 004324' 062701 173572          ADD     #DVREGS-.,R1
1152 004330' 012746 000005          MOV     #DVREGE-DVREGS/6,-(SP) ;GET # OF REGISTERS TO DISPLAY
1153 004334' 012167 000330          10$:  MOV     (R1)+,DVRGMG ;MOVE REG NAME TO MSG
1154 004340' 012167 000326          MOV     (R1)+,DVRGMG+2
1155 004344' 005721          TST    (R1)+
1156 004346' 012200          MOV     (R2)+,R0      ;BYPASS DISP VALUE
1157 004350' 010146          MOV     R1,-(SP)     ;GET REG'S STORED VALUE
1158 004352' 010246          MOV     R2,-(SP)     ;SAVE R1 & R2
1159 004354' 004577 173476          JSR    R5,@PINASC   ;CONVERT IT TO ASCII
1160 004360' 000316          .WORD  DVRGDT-
1161 004362' 004567 000044          JSR    R5,PRINT     ;PRINT THE STATUS MSG
1162 004366' 000302          .WORD  DVRGMG-
1163 004370' 000014          .WORD  12
1164 004372' 012602          MOV     (SP)+,R2     ;RESTORE R1 & R2
1165 004374' 012601          MOV     (SP)+,R1
1166 004376' 005316          DEC     (SP)
1167 004400' 001355          BNE    10$          ;DECR REG CNT
1168 004402' 005726          TST    (SP)+
1169 004404' 000205          RTS     R5          ;DONE ALL? (Y,N-10$)
                                     ;REMOVE COUNT FROM STACK
                                     ;EXIT IN-LINE

```



```

1171                                     ;PRINT CURRENT "BLK" VALUE
1172
1173                                     ;JSR   PC,PRTIWD      S/R CALL
1174                                     ;DESTROYS R0,R1,R2
1175
1176 004406' 016700 173372      PRTIWD: MOV   BLK,R0      ;GET BLK VALUE
1177 004412' 004577 173440      JSR   R5,JBINASC    ;CONVERT IT TO ASCII
1178 004416' 000550                                     .WORD  INFOBK-
1179 004420' 004567 000006      JSR   R5,PRINT      ;ISSUE BLOCK # MSG
1180 004424' 000535                                     .WORD  INFOMG-
1181 004426' 000013                                     .WORD  11.
1182 004430' 000207      RTS   PC      ;EXIT IN-LINE
1183
1184
1185                                     ;ISSUE MSG TO LIST DEVICE
1186
1187                                     ;JSR   R5,PRINT      S/R CALL
1188                                     ;.WORD  MSGADR-    REL ADR OF MSG
1189                                     ;.WORD  BYTCNT    MSG BYTE CNT (IF NEGATIVE,
1190                                     ;                RESET PRT DEV DEDICATED.)
1191                                     ;R3 = PROG TBL ADR
1192                                     ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
1193                                     ;DESTROYS R0,R1,R2
1194
1195 004432' 010500      PRINT:  MOV   R5,R0      ;GET MSG ADR & MAKE IT ABS
1196 004434' 062500      ADD   (R5)+,R0
1197 004436' 012501      MOV   (R5)+,R1      ;GET BYTE COUNT
1198 004440' 005704      TST   R4           ;USE CMND MODE PRINT?
1199 004442' 100030      BPL   40$         ;Y N-40$
1200 004444' 010702      MOV   PC,R2       ;SET UP LINK INFO ADR
1201 004446' 062702 000040  ADD   #20$-.,R2
1202 004452' 160200      SUB   R2,R0       ;MAKE MSG ADR REL
1203 004454' 010022      MOV   R0,(R2)+    ;STORE MSG ADR
1204 004456' 010112      MOV   R1,(R2)     ;STORE MSG'S BYTE COUNT
1205 004460' 100001      BPL   10$         ;CNT NEG? (Y,N-10$)
1206 004462' 005412      NEG   (R2)        ;MAKE IT POSITIVE
1207 004464' 016367 000006 000056 10$:  MOV   PASCIN(R3),PROGNM ;STORE PROG'S # IN MSG
1208 004472' 004577 173356      JSR   R5,ACLIST   ;ISSUE PROG #
1209 004476' 000050      .WORD  PNMMSG-
1210 004500' 000005      .WORD  5
1211 004502' 004577 173346      JSR   R5,ACLIST   ;ISSUE MSG SPECIFIED
1212 004506' 000000      20$: .WORD  XXXX
1213 004510' 000000      .WORD  XXXX
1214 004512' 004577 173336      JSR   R5,ACLIST   ;ISSUE A <CR> & <LF>
1215 004516' 000240      .WORD  CRLF-
1216 004520' 000002      .WORD  2
1217 004522' 000410      BR    PRTEX      ;GO TO EXIT
1218 004524' 010067 000010 40$:  MOV   R0,50$     ;STORE MSG'S ABS ADR
1219 004530' 010167 000006      MOV   R1,60$     ;STORE ITS BYTE CNT
1220 004534' 004577 173312      JSR   R5,ACLIST   ;GO TO MPG TO ISSUE THE MSG
1221 004540' 000000      50$: .WORD  XXXX
1222 004542' 000000      60$: .WORD  XXXX
1223 004544' 000205      PRTEX: RTS   R5      ;EXIT IN-LINE
    
```



```

1225 .SBTTL TC11 MESSAGE STORAGE AREA
1226
1227
1228 .NLIST BEX
1229
1230 .EVEN
1231 004546' 021520 PNMMSG: .ASCII /P# /
1232 004550' 054130 011 PROGMM: .ASCII /XX/<011>
1233 004553' 101 020124 040514 ATIMSG: .ASCII 'AT LAST INT/NON-INT TERM:'
1234 004604' 052503 051122 047105 CURMSG: .ASCII /CURRENTLY:/
1235 004616' 047105 020104 043117 RENDMG: .ASCII /END OF REPORT/
1236 .ODD
1237 004633' 052 025052 052052 UNITMG: .ASCII /***TC11 DECTAPE UNIT: /
1238 004662' 054130 054130 054130 UNASCI: .ASCII /XXXXXX/
1239 .EVEN
1240 004670' 054130 054130 020075 DVRGMG: .ASCII /XXXX= /
1241 004676' 054130 054130 054130 DVRGDT: .ASCII /XXXXXX/
1242 004704' 054502 042524 035123 CNTSMG: .ASCII /BYTES: RD= /
1243 004720' 054130 054130 054130 BCMRD: .ASCII /XXXXXXXXXXXXX WR= /
1244 004742' 054130 054130 054130 BCMWR: .ASCII /XXXXXXXXXXXXX/
1245 004756' 005015 CRLF: .ASCII <015><012>
1246 004760' 041411 047115 051504 .ASCII <011>/CMNDS: RD= /
1247 004775' 130 054130 054130 CMDCRD: .ASCII /XXXXXX WR= /
1248 005010' 054130 054130 054130 CMDCWR: .ASCII /XXXXXX MISC= /
1249 005025' 130 054130 054130 CMDCMS: .ASCII /XXXXXX/<015><012>
1250 005035' 011 051105 047522 .ASCII <011>/ERRORS: DEV= /
1251 005054' 054130 054130 054130 CNTERR: .ASCII /XXXXXX DATA= /
1252 005071' 130 054130 054130 CNTDER: .ASCII /XXXXXX/<015><012>
1253 005101' 011 047111 042524 .ASCII <011>/INTERRUPTS: /
1254 005117' 130 054130 054130 CNTINT: .ASCII /XXXXXX/
1255 005125' 005125' CNTSEN= .
1256 005125' 124 046511 047505 IOTO: .ASCII 'TIMEOUT ON I/O'
1257 005143' 102 045514 051440 BSCHER: .ASCII /BLK SEARCH ERR/
1258 005161' 102 045514 020075 INFOMG: .ASCII /BLK= /
1259 005166' 054130 054130 054130 INFOBK: .ASCII /XXXXXX/
1260 .EVEN
1261 005174' 052123 047115 020124 STMNMG: .ASCII /STMNT # /
1262 005204' 054130 054130 054130 STMNUM: .ASCII /XXXXXX/
1263 005212' 052123 052101 051525 TCEMSG: .ASCII /STATUS ERROR: /
1264 005230' 000023 CODFLD: .BLKB 19.
1265 005253' 111 053116 052440 INV DVN: .ASCII /INV UNIT #/
1266 005265' 111 053116 041040 INVBKN: .ASCII /INV BLK #/
1267 .EVEN
1268
1269 .LIST BEX
1270
1271 005276' DVREND= .

```



```

1273          .SBTTL FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES
1274
1275          ; PROGRAM TABLE FORMAT
1276
1277          000242 PTLGTH= 162. ;PROGRAM TABLE LENGTH - NON MEM MGMNT VERSION OF MPG
1278
1279          ;(PTLGTH= 212. ;PROGRAM TABLE LENGTH - MEM MGMNT VERSION OF MPG)
1280
1281          000000 PFLGWD= +0. ;PROGRAM FLAG WORD - 1 WORD
1282
1283          000002 URSTOP= 2 ; 1 = USER HAS STOPPED THIS PROGRAM
1284          000004 ERSTOP= 4 ; 1 = AN ERROR HAS STOPPED THIS PROGRAM
1285          000010 WT4IOT= 10 ; 1 = WAITING FOR I/O TERMINATION
1286          000020 CTPRIO= 20 ; 1 = CONSOLE OR PRINTER I/O IN PROGRESS
1287          000040 SETDED= 40 ; 1 = THIS PROG SET THE PRT DEV DEDICATED FLAG
1288          000100 OCPRES= 100 ; 1 = OBJ CODE IS PRESENT
1289          000200 USEUBM= 200 ; 1 = THIS PROG USES THE UNIBUS MAP (MEM MGMNT ONLY)
1290          100000 ACTIVE= 100000 ; 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)
1291
1292          000002 POPSW= +2. ;PROGRAM'S OPERATION SWITCHES - 1 WORD
1293
1294          100000 STONER= 100000 ; 1 = STOP PROG EXECUTION UPON ERROR
1295          040000 CYCPRG= 40000 ; 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
1296          020000 PRONER= 20000 ; 1 = DO NOT PRINT ON ERROR
1297          010000 BIT12= 10000 ; 0 = NOT USED
1298          004000 BIT11= 4000 ; 0 = NOT USED
1299          002000 CYCDVL= 2000 ; 1 = CYCLE THE DEVICE LIST
1300          001000 GTNXTD= 1000 ; 1 = CYCLE ON SAME DEVICE UPON ERROR
1301          000400 DOERCK= 400 ; 1 = DON'T DO ERROR CHECKING
1302          000200 SPOPER= 200 ; 1 = DEVICE SPECIAL OPERATION
1303          000100 BIT6= 100 ; 0 = NOT USED
1304          000040 DOIOT= 40 ; 1 = DO NOT PERFORM I/O TIMEOUT
1305          000020 AUTORP= 20 ; 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
1306          000010 AURPEP= 10 ; 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
1307          000004 HSKPEP= 4 ; 1 = HOUSEKEEP COUNTS ONLY AT RUN COMMAND
1308          000002 PFBBOV= 2 ; 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
1309          000001 NOCOMP= 1 ; 1 = DO NOT PRINT PROG COMPLETED MSG
1310
1311          000004 PFWADR= +4. ;*;PROGRAM FLAGWORD ADDRESS - 1 WORD
1312
1313          000006 PASCIN= +6. ;PROGRAM'S NUMBER IN ASCII - 1 WORD
1314
1315          000010 PNAME= +8. ;PROGRAM'S NAME IN ASCII - 6 BYTES
1316
1317          000016 PRDIOA= +14. ;ADDRESS OF READ I/O AREA - 1 WORD
1318
1319          000020 PWRIOA= +16. ;ADDRESS OF WRITE I/O AREA - 1 WORD
1320
1321          000022 PSRCST= +18. ;SOURCE STATEMENTS START ADDRESS - 1 WORD
1322
1323          000024 POBJST= +20. ;OBJECT CODE START ADDRESS - 1 WORD
1324
1325          000026 PLNGTH= +22. ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD
1326
1327          000030 PTOCNT= +24. ;I/O TIMEOUT COUNT - 1 WORD
1328
    
```



## E03

MAINDEC-11-DTTCA-B TC11/TU56 DEVICE ROUTINE FOR MPG MACY11 27(732) 24-SEP-76 14:00 PAGE 12-1  
 DTTTAB.P11 FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

SEQ 0383

1329	000032	PMDLCD= +26.	;DEV ROUT MODEL # CODE - 1 WORD
1330			
1331	000034	PDPNTR= +28.	;CURRENT DEVICE NUMBER POINTER - 1 BYTE
1332			
1333	000035	PCURDV= +29.	;CURRENT DEVICE # - 1 BYTE
1334			
1335	000036	PDNUMS= +30.	;DEVICE NUMBERS - 16 BYTES
1336			
1337	000056	PTEM0= +46.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1338			
1339	000060	PTEM1= +48.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1340			
1341	000062	PTEM2= +50.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1342			
1343	000064	PTEM3= +52.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1344			
1345	000066	PTEM4= +54.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1346			
1347	000070	PTEM5= +56.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1348			
1349	000072	PTEM6= +58.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1350			
1351	000074	PTEM7= +60.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1352			
1353	000076	PTEM8= +62.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1354			
1355	000100	PTEM9= +64.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1356			
1357	000102	PTEM10= +66.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1358			
1359	000104	PTEM11= +68.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1360			
1361	000106	PTEM12= +70.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1362			
1363	000110	PTEM13= +72.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1364			
1365	000112	PTEM14= +74.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1366			
1367	000114	PTEM15= +76.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1368			
1369	000116	PNBR= +78.	;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
1370			
1371	000120	PSRC= +80.	;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
1372			
1373	000122	PDST= +82.	;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
1374			
1375	000124	PSTKCT= +84.	;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
1376			
1377	000126	PSTKSV= +86.	;STACK WORDS STORAGE AREA - 30 WORDS
1378			
1379	000222	PSVREG= +146.	;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
1380			
1381	000236	PUSRPC= +158.	;USER'S CURRENT PROGRAM COUNTER - 1 WORD
1382			



```

1384           ;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMNT VERSION
1385
1386           ;(PRDIOX= +160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
1387           ;(PRDIOV= +164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
1388           ;(PWRIOX= +168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
1389           ;(PWRIOV= +172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
1390           ;(PUPARS= +176. ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)
1391           ;(PUPDRS= +192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
1392           ;(PUBMAP= +208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)
1393
1394           ;END OF MEM MGMNT ONLY ENTRIES
1395
1400           000240      PTSIZE= +160. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMNT
1401
1402           ;(PTSIZE= +210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMNT VERSION)
1403
1404           000242      PTEND= +162. ;END OF PROGRAM TABLE - NON MEM MGMNT VERSION
1405
1406           ;(PTEND= +212. ;END OF PROGRAM TABLE - MEM MGMNT VERSION)
1407
1408

```



```

1410 ; DEVICE ROUTINE TABLE
1411
1412
1413 000116 DRTLTH= 78. ;DEVICE ROUTINE TABLE LENGTH
1414 ;
1415 ;
1416 000000 DEVRSZ= +0. ;DEVICE ROUTINE SIZE IN BYTES - 1 WORD
1417 ;
1418 000002 DEVFWD= +2. ;DEVICE ROUTINE FLAGWORD - 1 WORD
1419 ;
1420 000004 DEVIW1= +4. ;DEVICE INTERFACE WORD # 1 - 1 WORD
1421 ;
1422 000006 DEVIW2= +6. ;DEVICE INTERFACE WORD # 2 - 1 WORD
1423 ;
1424 000010 DEVIW3= +8. ;DEVICE INTERFACE WORD # 3 - 1 WORD
1425 ;
1426 000012 DEVIW4= +10. ;DEVICE INTERFACE WORD # 4 - 1 WORD
1427 ;
1428 000014 DEVIW5= +12. ;DEVICE INTERFACE WORD # 5 - 1 WORD
1429 ;
1430 000016 DEVIW6= +14. ;DEVICE INTERFACE WORD # 6 - 1 WORD
1431 ;
1432 000020 DEVIW7= +16. ;DEVICE INTERFACE WORD # 7 - 1 WORD (SIZE)
1433 ;
1434 000022 DEVIW8= +18. ;DEVICE INTERFACE WORD # 8 - 1 WORD (ERR)
1435 ;
1436 000024 DEVDRA= +20. ;DEVICE REGISTERS ADDRESS - 1 WORD
1437 ;
1438 000026 DEVIVA= +22. ;DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
1439 ;
1440 000030 DEVRPS= +24. ;DEVICE READ PROCESSOR STATUS WORD (BUS REQ) - 1 WORD
1441 ;
1442 000032 DEWPS= +26. ;DEVICE WRITE PROC STATUS WORD (BUS REQ) - 1 WORD
1443 ;
1444 000034 DHKPAD= +28. ;DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
1445 ;
1446 000036 DERPAD= +30. ;DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
1447 ;
1448 000040 DKILAD= +32. ;DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
1449 ;
1450 000042 DECTAD= +34. ;DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
1451 ;
1452 000044 DTOEAD= +36. ;DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
1453 ;
1454 000046 DEVI0B= +38. ;DEVICE I/O BUSY BRANCH ADDRESS (CIOBSY) - 1 WORD
1455 ;
1456 000050 DEVDER= +40. ;DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
1457 ;
1458 000052 DVUPRT= +42. ;USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
1459 ;
1460 000054 DVCprt= +44. ;CMND MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
1461 ;
1462 000056 DEVBTA= +46. ;CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
1463 ;
1464 000060 DVBTDA= +48. ;CONVERT BINARY TO DECIMAL ASCII BR ADR (BTASLZ) - 1 WORD
1465

```



# H03

MAINDEC-11-DTTCA-B TC11/TU56 DEVICE ROUTINE FOR MPG  
DTTCAB.P11 FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

MACY11 27(732) 24-SEP-76 14:00 PAGE 12-4

SEQ 0386

1466	000062	DVPDTA= +50.	; CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
1467			
1468	000064	DVSFWD= +52.	; MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
1469			
1470	000066	DVSVEC= +54.	; SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
1471			
1472	000070	DVCVEC= +56.	; CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
1473			
1474	000072	DVTVEC= +58.	; TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
1475			
1476	000074	DVRINT= +60.	; RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
1477			
1478	000076	DVGETB= +62.	; GET DATA BYTE BR ADR (GETBYT) - 1 WORD
1479			
1480	000100	DVPUTB= +64.	; PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
1481			
1482	000102	DEVSTP= +66.	; DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
1483			
1484	000104	DEVETP= +68.	; DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
1485			
1486	000106	DVPTEP= +70.	; PACK TABLE EXTEN. REL POINTER - 1 WORD
1487			
1488	000110	DVVTEP= +72.	; VECTOR TABLE EXTEN. REL POINTER - 1 WORD
1489			
1490	000112	DVCTEP= +74.	; COMPILER TBL EXTEN. REL POINTER - 1 WORD
1491			
1492	000114	DVIWSP= +76.	; DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
1493			
1494	000116	DRTEND= +78.	; END OF DEVICE ROUTINE TABLE
1495			
1496			
1497	000001	.END	



ACTIVE= 100000		DEVIVA= 000026		DVUPRT= 000052		NONINT 002226R	002	PUTBYT 000100R	002
ATIMSG 004553R	002	DEVIW1= 000004		DVVTEP= 000110		NOWAIT 001466R	002	PWRIOA= 000020	
AURPEP= 000010		DEVIW2= 000006		ERCDTB 003776R	002	OCPRES= 000100		RDALL 001610R	002
AUTORP= 000020		DEVIW3= 000010		ERMBAS 003352R	002	PASCIN= 000006		RDCNT 000706R	002
BCMRD 004720R	002	DEVIW4= 000012		ERP8KN 003404R	002	PC =%000007		RDCOM 001526R	002
BCMWR 004742R	002	DEVIW5= 000014		ERR 000022R	002	PCURDV= 000035		RDNUM 001576R	002
BINASC 000056R	002	DEVIW6= 000016		ERRADR 000724R	002	PDNUMS= 000036		READ 001516R	002
BIT11 = 004000		DEVIW7= 000020		ERRCNT 000714R	002	PDPNTR= 000034		RENDMG 004616R	002
BIT12 = 010000		DEVIW8= 000022		ERRCOM 003312R	002	PDST = 000122		REPORT 001000R	002
BIT6 = 000100		DEVVPS= 000030		ERRCS 003266R	002	PFBBOV= 000002		REPTBL 001274R	002
BLK 000004R	002	DEVRSZ= 000000		ERRCS1 003274R	002	PFLGWD= 000000		RESREG 004140R	002
BSCHER 005143R	002	DEVSTP= 000102		ERREX 003466R	002	PFWADR= 000004		REV 001506R	002
BTASLZ 000060R	002	DEVWPS= 000032		ERRIS 003304R	002	PLNGTH= 000026		REVCNT 000742R	002
BYRD 000676R	002	DFLGWD 000002R	002	ERRSNM 003410R	002	PMDLCD= 000032		RINTEX 004072R	002
BYWR 000702R	002	DHKPAD= 000034		ERSTAD 003402R	002	PNAME = 000010		RINTV 004050R	002
CI OBSY 000046R	002	DISCNT 001160R	002	ERSTOP= 000004		PNBR = 000116		RPTBAS 001230R	002
CKDBSY 003166R	002	DISCT1 001164R	002	FINCNT 000740R	002	PNMMSG 004546R	002	RPTEND 001254R	002
CLIST 000054R	002	DISPST 004316R	002	FUNCEX 001474R	002	POBJST= 000024		RPTLP 001212R	002
CLRVEC 000070R	002	DISUIV 004264R	002	FWD 001476R	002	POPSW = 000002		RTNINT 000074R	002
CMDCMS 005025R	002	DISUML 004312R	002	GETBYT 000076R	002	PRDIOA= 000016		R0 =%000000	
CMDCOM 001702R	002	DISUNM 004226R	002	GTNXTD= 001000		PRINT 004432R	002	R1 =%000001	
CMDCRD 004775R	002	DISUPR 004304R	002	HSKEEP 000744R	002	PROCTM 003470R	002	R2 =%000002	
CMDCWR 005010R	002	DKILAD= 000040		HSKPEN= 000744R	002	PROGNM 004550R	002	R3 =%000003	
CMDEND 002212R	002	DOERCK= 000400		HSKPEP= 000004		PRONER= 020000		R4 =%000004	
CMDEX 002216R	002	DOIOT = 000040		HSKPST= 000652R	002	PRTEX 004544R	002	R5 =%000005	
CNTADR 000726R	002	DREGAD 000024R	002	INBLKN 000736R	002	PRTIWD 004406R	002	SAVREG 004124R	002
CNTDER 005071R	002	DRTEND= 000116		INFOBK 005166R	002	PSRC = 000120		SEARCH 002732R	002
CNTERR 005054R	002	DRTLTH= 000116		INFOMG 005161R	002	PSRCST= 000022		SETDED= 000040	
CNTINT 005117R	002	DTOEAD= 000044		INTCNT 000720R	002	PSTKCT= 000124		SETVEC 000066R	002
CNTSEN= 005125R	002	DVBTDA= 000060		INVBKN 005265R	002	PSTKSV= 000126		SIZE 000020R	002
CNTSMG 004704R	002	DVCMS 000154R	002	INVVDN 005253R	002	PSVREG= 000222		SP =%000006	
CODFLD 005230R	002	DVCPT= 000054		IOTO 005125R	002	PSWD 000030R	002	SPOPER= 000200	
CRLF 004756R	002	DVCPT= 000466R	002	ISTAT = 000652R	002	PTEMO = 000056		STMNMG 005174R	002
CSTAT 000664R	002	DVCTEP= 000112		IVCTAD 000026R	002	PTEM1 = 000060		STMNUM 005204R	002
CSYSFW 000064R	002	DVCVEC= 000070		KILL 001420R	002	PTEM10= 000102		STONER= 100000	
CTPRIO= 000020		DVGETB= 000076		KILLEX 001444R	002	PTEM11= 000104		STOP 001646R	002
CUPGER 000050R	002	DVIWSP= 000114		LCOUNT 000632R	002	PTEM12= 000106		STPALL 001670R	002
CURCMD 000734R	002	DVIWST 000622R	002	LFWD 000632R	002	PTEM13= 000110		STSTAT 004202R	002
CURCNT 000732R	002	DVMVTE 000406R	002	LNWAIT 000632R	002	PTEM14= 000112		SUPTAD 004156R	002
CURFLG 000730R	002	DVPDTA= 000062		LOCZ 000000R	002	PTEM15= 000114		TCMSG 005212R	002
CURMSG 004604R	002	DVPKTE 000246R	002	LRDALL 000633R	002	PTEM2 = 000062		TCINT 002536R	002
CYCDVL= 002000		DVPTEP= 000106		LRDNUM 000633R	002	PTEM3 = 000064		TOECNT 000722R	002
CYCPRG= 040000		DVPUTB= 000100		LREV 000632R	002	PTEM4 = 000066		TOUTER 001320R	002
DATAER 000716R	002	DVREGE= 000154R	002	LSTALL 000632R	002	PTEM5 = 000070		TOUTEX 001416R	002
DECASC 000062R	002	DVREGS 000116R	002	LSTATS 000632R	002	PTEM6 = 000072		TSTVEC 000072R	002
DECTAD= 000042		DVREND= 005276R	002	LSTOP 000632R	002	PTEM7 = 000074		TVECT 004074R	002
DERPAD= 000036		DVREX 001264R	002	LWAIT 000632R	002	PTEM8 = 000076		TVECTX 004122R	002
DEVBTA= 000056		DVRGDT 004676R	002	LWRALL 000642R	002	PTEM9 = 000100		ULIST 000052R	002
DEVDER= 000050		DVRGMG 004670R	002	LWRM 000642R	002	PTEND = 000242		UNASCI 004662R	002
DEVORA= 000024		DVRINT= 000074		MISCNT 000712R	002	PTLGTH= 000242		UNITMG 004633R	002
DEVETP= 000104		DVSFWD= 000064		MISCOM 001656R	002	PTCNT= 000030		URSTOP= 000002	
DEVFWD= 000002		DVSVEC= 000066		NINTSU 002516R	002	PTSIZE= 000240		USEUBM= 000200	
DEVI0B= 000046		DVTVEC= 000072		NOCOMP= 000001		PUSRPC= 000236		WAIT 001446R	002



J03

MAINDEC-11-DTTCAB-B TC11/TU56 DEVICE ROUTINE FOR MPG MACY11 27(732) 24-SEP-76 14:00 PAGE 13-1  
DTTCAB.P11 SYMBOL TABLE

SEQ 0388

WRALL 001622R 002 WRCOM 001556R 002 WRTM 001634R 002 WT4IOT= 000010 . = 005276R 002  
. ABS. 000000 000  
000000 001  
TC11 005276 002

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

\*.DTTCAB/NL:TOC/DOC=DTTCAB.P11  
RUN-TIME: 49.9 SECONDS  
RUN-TIME RATIO: 171/14=11.5  
CORE USED: 5K (9 PAGES)

DOCUMENT PAGES: 35



