

TR79F

TR79F UTILITY TIMING
MD-11-DZRTB-B

EP-DZTRB-B-DL-B
COPYRIGHT 1977
FICHE 1 OF 1

MAR 1977
digital
MADE IN USA

The left side of the page contains a vertical column of technical information. It consists of approximately 15 small, rectangular blocks stacked vertically. Each block appears to contain a small table or a diagram with some text. The text is too faint to read, but the layout suggests a structured list of data points or parameters. The blocks are separated by thin horizontal lines.

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTRB-B-D
 PRODUCT NAME: TR79 UTILITY PROGRAMS
 DATE: JANUARY 1977
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHOR: JOHN YASKO TELCO PRODUCT SUPPORT (EX 3922)

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1976, 1977
 DIGITAL EQUIPMENT CORPORATION, MAYNARD MASS.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

CO1

TR79 UTILITY PROGRAM

TABLE OF CONTENTS

SECTION	SUBJECT
1.	ABSTRACT
2.	REQUIREMENTS

44-38861-1038
44-38861-1037
44-38861-1036
44-38861-1035
44-38861-1034
44-38861-1033
44-38861-1032
44-38861-1031
44-38861-1030
44-38861-1029
44-38861-1028
44-38861-1027
44-38861-1026
44-38861-1025
44-38861-1024
44-38861-1023
44-38861-1022
44-38861-1021
44-38861-1020
44-38861-1019
44-38861-1018
44-38861-1017
44-38861-1016
44-38861-1015
44-38861-1014
44-38861-1013
44-38861-1012
44-38861-1011
44-38861-1010
44-38861-1009
44-38861-1008
44-38861-1007
44-38861-1006
44-38861-1005
44-38861-1004
44-38861-1003
44-38861-1002
44-38861-1001

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

- 3. LOADING PROCEDURE
- 4. STARTING PROCEDURE
- 5. RESTRICTIONS
- 6. CONSOLE SETTINGS
- 7. OPERATION
- 8. LISTING

1. ABSTRACT

THIS PROGRAM IS IN TWO PARTS, AND IS INTENDED TO PROVIDE THE USER WITH A TOOL FOR TROUBLE-SHOOTING THE TR79 MAGTAPE SUBSYSTEM ON A PDP-11 COMPUTER SYSTEM. THE FIRST PART OF THE PROGRAM ALLOWS THE USER TO GIVE THE MAGTAPE, COMMANDS, TO SIMULATE USER ROUTINES BY MERELY INSERTING THESE COMMANDS IN THE CORE LOCATIONS PROVIDED. THE USER MAY EXECUTE ONE OR SEVERAL INSTRUCTIONS IN ANY LEGAL SEQUENCE. WHILE THE CODE FOR THE DRIVER IS SIMPLE AND USES NO INTERRUPTS, DUE TO THE DESIGN OF THE HARDWARE CERTAIN ERROR CONDITIONS MUST BE IDENTIFIED IN ORDER TO PREVENT MISINTERPRITATION OF THE DESIRED RESULTS.

PART TWO OF THE PROGRAM CONSIST OF SELF CONTAINED ROUTINES TO PERMIT THE USER TO SET UP AND CHECK THE DELAYS CONTAINED WITHIN THE TR79 CONTROLLER, BY USING THE SWITCH REGISTER TO SELECT THE APPROPRIATE ROUTINE.

2. REQUIREMENTS

2.1 HARDWARE

- A. PDP-11 PROCESSOR
- B. TR79 MAGTAPE TRANSPORT (HP-7970E DRIVE)
- C. TR79F MAGTAPE CONTROLLER

2.2 STORAGE

THIS PROGRAM REQUIRES A MINIMUM OF 4K OF CORE

113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168

3. LOADING

USE STANDARD BINARY LOADING PROCEDURE

4.0 STARTING PROCEDURE

THERE ARE TWO STARTING ADDRESSES THAT MAY BE USED

4.1 200 (8): LA 200 SR=0 A START AT THIS ADDRESS WILL RESULT IN A PROGRAMMED DEFAULT OPERATION OF A WRITE FORWARD WITH A WORD COUNT OF -20 AND A DATA PATTERN OF ALL 1'S. TO MODIFY THESE PARAMETERS SEE SECTION 7.1 PROGRAM OPERATION NOTE: ALSO SEE SECTION 5.0 PROGRAM RESTRICTIONS, THE DEFAULT OF WRITE WILL NOT WORK IF TAPE IS AT B.O.T..

4.2 204 (8) LA 204 SR=0 A START AT THIS ADDRESS WILL EXECUTE THE SPECIALLY DESIGNED SETUP ROUTINES TO ALLOW THE USER TO SETUP OR VERIFY THE DELAYS WITHIN THE TR79 CONTROLLER.
NOTE: ALWAYS USE SCRATCH TAPES WHEN TAPE MOTION IS INDICATED.

5.0 RESTRICTIONS

5.1 A. A PSEUDO-OP OF A 20(8) HAS BEEN PROVIDED TO ALLOW THE USER TO POWER CLEAR BETWEEN OPERATIONS IF DESIRED, HOWEVER THE PROGRAM CAN RECOVER FROM THIS IS A POWER CLEAR AND TAKES 900 MILI SECONDS TO COMPLETE ANY ATTEMPTS TO ISSUE INSTRUCTIONS TO THE CONTROLLER WHILE A POWER CLEAR IS IN PROGRESS WILL RESULT IN ILLEGAL COMMAND BIT SETTING WHICH WILL INHIBIT ANY FURTHER INSTRUCTIONS FROM BEING EXECUTED. A POWER CLEAR IS ALSO GENERATED FROM A BUS INIT WHICH OCCURS FROM A RESET INSTRUCTION, THE DRIVER USES NO RESETS, (USE CAUTION IF YOU MODIFY THE DRIVER PACKAGE)

B. THE TR79 CONTROLLER CHECKS FOR CERTAIN ILLEGAL FUNCTIONS DUE TO TAPE POSITION OR STATUS, THE DRIVER PACKAGE WILL CHECK THESE CONDITIONS AND HALT AT APPROPRIATE LOCATIONS WITH MEANINGFULL DATA DISPLAYED (SEE SECTION 7.2 ERROR CHECKS).
THE LISTED CONDITIONS WILL PRODUCE ILLEGAL COMMAND ERRORS:

1. ATTEMPT TO WRITE DATA FROM LOAD POINT WITHOUT AN I.D.B.
2. ATTEMPT TO WRITE A TAPE MARK FROM LOAD POINT
3. ATTEMPT TO MOVE TAPE IN REVERSE FROM LOAD POINT
4. ATTEMPT TO REWIND FROM LOAD POINT

169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224

5. ATTEMPT TO WRITE AN I.D.B. AT OTHER THAN LOAD POINT
6. ATTEMPT TO WRITE DATA WITH THE WRITE RING REMOVED
7. ISSUE A COMMAND WHILE THE MAGTAPE IS NOT READY
8. ISSUE A COMMAND WHILE THE CONTROLLER IS NOT READY
9. ISSUE A COMMAND WITH INHIBIT BIT SET
10. ILLEGAL FUNCTION CODES 00,03,05,06,11,12,14

C. THE PROGRAM DOES NO DATA CHECKS ON READ OR WRITE DATA TRANSFERRED. IT IS THE RESPONSIBILITY OF THE OPERATOR TO MANUALLY EXAMINE THE BUFFER LOCATIONS TO DETERMINE IF THERE HAVE BEEN ANY PICKED OR DROPPED BITS IF DESIRED.

D. NOTE: HARDWARE OPERATION OF THE TR-79 SPECIFIES THAT EACH CORE WORD LOCATION CONTAIN ONE BYTE (BITS 0-7) OF DATA AND PARITY (BIT 8). THEREFORE WHEN CALCULATING THE WORD COUNT FOR A TRANSFER THE ACTUAL NUMBER OF CORE LOCATIONS ACCESED IS EQUAL TO 2X THE NUMBER LOADED IN THE WORD COUNT REGISTER. ALSO NOTE THAT THE CONTROLLER DOES NOT APPEND PARITY TO THE BYTE BEFORE DOING A WRITE OPERATION. PARITY MUST BE CORRECT IN CORE OTHERWISE ERRORS WILL OCCUR ON THE TRANSFER (CDD PARITY) IS ALW

6.0 CONSOLE SWITCH SETTINGS

- SW 15 = 1 STOP AFTER EACH OPERATION (ONLY WITH START 200)
0 PROCEED
- SW 14 = 1 STOP AT THE END OF EACH PROGRAM PASS (ONLY WITH START 200)
0 PROCEED
- SW 7 = 1 ENABLE FOR DELAY ROUTINES (EXECUTE ROUTINE ONLY WITH START 204)
0 ALLOW SELECTION OF DELAY ROUTINES WITH SW 0-3
- SW 0 THU 3 = DELAY ROUTINE TO BE EXECUTED (ONLY WITH START 204)

6.1 DELAY SETUP TABLE

SWITCH SETING	DELAY NAME	MODULE TYPE	LOCATION	PRINT PAGE	INPUT PIN	OUTPUT PIN	TIME
00	NO-OP						
01	P CLR	M-302	C-06	T02-2	H2	F2	20 MILI SEC.
02	P CLR OFF	M-306	D-09	T04-1	H2	T2	900 MILI SEC.
03	ERROR CLK	M-302	C-10	T04-2	H2	F2	200 NANO SEC.

225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280

04	I	WRITE ENAB	I	M-302	I	C-10	I	T09-3	I	M2	I	T2	I	40 MICRO SEC.	I
05	I	BUFF CONT	I	M-304	I	B-18	I	T11-1	I	E1	I	J1	I	1 MICRO SEC.	I
06	I	END WR DAT	I	M-302	I	A-16	I	T11-1	I	M2	I	T2	I	18 MICRO SEC.	I
07	I	1ST WD REQ	I	M-302	I	A-22	I	T11-2	I	H2	I	F2	I	100 MICRO SEC.	I
10	I	ERASE	I	M-304	I	B-18	I	T09-3	I	S1	I	M1	I	1 MICRO SEC.	I
11	I	WRITE IDB	I	M-302	I	A-16	I	T09-1	I	H2	I	F2	I	17 MILI SEC.	I
12	I	IDB TIMING	I	M-302	I	D-13	I	T09-1	I	H2	I	F2	I	75 MILI SEC.	I
13	I	ABORT	I	M-306	I	A-25	I	T09-3	I	H2	I	T2	I	1.5 SEC.	I
14	I	BUSY DELAY	I	M-304	I	B-18	I	T05-1	I	R1	I	P1	I	100 NANO SEC.	I
15	I	GO BIT DEL	I	M-304	I	B-18	I	T06-1	I	D1	I	H1	I	1 MICRO SEC.	I
16	I	M.S.D.	I	M-302	I	A-22	I	T09-2	I	M2	I	T2	I	900 MILI SEC.	I
17	I	NO-OP	I		I		I		I		I		I		I

 * DELAY CONDITION NOTES *

6.2 DELAY

- 00 NO OPERATION PERFORMED WAITING SWITCH SELECTION AND ENABLE
- 01 POWER CLEAR PERFORMED NO TAPE MOTION DELAY PULSE IS POSITIVE GOING
- 02 POWER CLEAR PERFORMED NO TAPE MOTION DELAY PULSE IS POSITIVE GOING
- 03 NO TAPE MOTION, PROGRAM FORCES AN ERROR WITH THE BGL BIT IN THE TR STATUS REGISTER (BIT 11) DELAY PULSE IS POSITIVE GOING
- 04 TAPE MOTION, PROGRAM DOES A SHORT ERASE WHILE MOVING TAPE TAPE MOTION IS NOT READILY NOTICIBLE WHILE EXECUTING THIS ROUTINE DELAY PULSE IS POSITIVE GOING
- 05 TAPE MOTION, PROGRAM DOES A 10 BYTE WRITE, PROGRAM CHECKS FOR LOAD POINT AND WILL WRITE AN I.D.B. BEFORE ENTERING THE DELAY LOOP. DELAY PULSE IS POSITIVE GOING
- 06 SAME CONDITIONS AS 05 DELAY PULSE IS POSITIVE GOING
- 07 SAME CONDITIONS AS 05 DELAY PULSE IS POSITIVE GOING

- 281 10 TAPE MOTION, PROGRAM WILL CHECK FOR LOAD POINT THEN DO A
- 282 MAXIMUM ERASE TO MAKE THE OPERATION CONTINUOUS THE PROGRAM
- 283 WILL CLEAR THE ERASE COUNT BEFORE THE OPERATION IS DONE.
- 284 DELAY PULSE IS NEGATIVE GOING
- 285
- 286 11 TAPE MOTION, PROGRAM WILL CONTINUOUSLY WRITE THE I.D.B.
- 287 DELAY PULSE IS POSITIVE GOING
- 288
- 289 12 SAME CONDITIONS AS DELAY 11. DELAY PULSE IS POSITIVE GOING
- 290
- 291 13 PROGRAM WILL REWIND TAPE TO L.P. AND FORCE AN ERROR BY DOING
- 292 A WRITE DATA. DELAY PULSE IS POSITIVE GOING.
- 293
- 294 14 TAPE WILL MOVE TO L.P. , AND DO A MAXIMUM ERASE.
- 295 WHILE THIS IS HAPPENING PROGRAM WILL LOAD THE COMMAND
- 296 REGISTER TO PRODUCE A LD CTRL PULSE. DELAY PULSE IS POSITIVE GOING
- 297
- 298 15 SAME CONDITIONS AS DELAY 04. DELAY PULSE IS NEGATIVE GOING
- 299
- 300 16 PROGRAM WILL MOVE TAPE TO E.O.T. AND ATTEMPT TO DO A FAST
- 301 FORWARD TO PRODUCE THE MOTION STOP DELAY. DELAY IS POSITIVE.
- 302 NOTE: AFTER COMPLETION OF THIS ROUTINE A MANUAL REWIND
- 303 SHOULD BE PERFORMED.
- 304
- 305 17 THIS IS A NO OPERATION SAME AS 00
- 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

7.0 OPERATION

THE PROGRAM IS QUITE SIMPLE HOWEVER IT DOES REQUIRE KNOWLEGE OF THE OF THE TR-79 MAGTAPE SYSTEM AND AN UNDERSTANDING OF THE PROGRAM FUNCTIONS AND RESTRICTIONS. THE CODE HAS BEEN ASSEMBLED IN IMMEDIATE AND ABSOLUTE MODES USING PC ADRESSING. IT IS RECOMMENDED THAT THE USER READ AND UNDERSTAND THE RESTRICTIONS AND OPERATIONS SECTIONS.

THE DRIVER PROGRAM (LOAD ADRESS 200 START SWITCHES =0) CAN BE MADEB TO EXECUTE ANY LEGAL SEQUENCE OF OPERATIONS (SEE SECTION 7.3) BY INSERTING THE COMMANDS IN THE OPERATIONS TABLE, (CORE LOCATIONS 722 THRU 766). EACH COMMAND SHOULD OCCUPY ONE CORE LOCATION BITS 0-4 ONLY. THE TOTAL NUMBER OF COMMANDS TO BE EXECUTED SHOULD THEN BE ENTERED IN LOCATION 720. THE PROGRAM PARAMETERS MAY BE ALTERED BY CHANGING THE APPROPRIATE CORE LOCATIONS (SEE SECTION 7.1). PROGRAM DEFAULT IS A SINGLE WRITE COMMAND OF 20 WORDS OF ALL 1'S FROM LOCATION 2700 WITH MINIMUM DELAY BETWEEN OPERATIONS. THIS DEFAULT WILL NOT WORK IF THE TAPE IS POSITIONED AT LOAD POINT.

THE DELAY PROGRAM (LOAD 204 START SWITCHES=0) WILL EXECUTE THE DELAY SET-UP ROUTINES TO ALLOW SET-UP OF ALL THE DELAYS IN THE TR-79 CONTROLLER THE PROGRAM HAS AN ACTIVE SWITCH REGISTER AFTER STARTING. BY SELECTING THE DESIRED DELAY ROUTINE IN SWITCH REGISTER 0 THRU 3, AND THEN SETTING BIT 7 =1 THE ROUTINE WILL BEGIN EXECUTION. TO CHANGE THE DELAY ROUTINE SET BIT 7=0, WAIT A FEW SECONDS FOR COMPLETION OF THE ROUTINE, THEN ENTER THE NEW ROUTINE NUMBER IN BITS 0-3 AND SET BIT 7=1. THE DELAY PROGRAM CONTAINS NO ERROR HALTS, HOWEVER IF ERRORS ARE DETECTED THE

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
379
380
381
382
383
384
385
386
387
388
389
390
391
392

PROGRAM WILL INFORM THE USER BY OUTPUTTING A BELL CODE TO THE CONSOLE TERMINAL. THE PROGRAM WILL THEN DO A CONTROL RESET AND CONTINUE.

NOTE: THE PROGRAM BUILDS THE CORE DATA BUFFERS EACH TIME THE PROGRAM IS STARTED. THE PROGRAM DEFAULT IS LOCATION 2700 HOWEVER THIS MAY BE CHANGED BY MODIFYING LOCATION 242 IN THE CORE BUILD ROUTINE TO PUT THE BUFFERS ANYPLACE IN THE LOWER 28K. THIS PROGRAM DOES NOT PROGRAM THE KT AND DOES NOT RELOCATE ABOVE THE LOWER 28K OF MEMORY.

7.1 PROGRAM PARAMETER LIST AND CORE ADRESSES

PARAMETER	LOCATION	DESCRIPTION
EXTENDED CORE ADDRESS	700	BITS 12 AND 13 OF THIS LOCATION REPRESENT XBA 16 AND XBA 17 OF THE TR CONTROL REGISTER THESE BITS ALLOW RELOCATION OF THE DATA BUFFER.
UNIT SELECT	702	BITS 8+9 IN THIS LOCATION REPRESENT THE UNIT NUMBERS OF THE TAPE DRIVES. A MAXIMUM OF 4 DRIVES PER CONTROLLER DEFAULT IS UNIT 0.
WORD COUNT	704	THIS IS THE 2'S COMPLIMENT OF THE NUMBER OF WORDS TRANSFERRED. SINCE EACH BYTE OCCUPIES A WORD LOCATION THE NUMBER OF CORE LOCATIONS USED IS 2X THE WORD COUNT. PROGRAM DEFAULT IS -20 WORDS.
READ ADRESS	706	CONTAINS ADRESS OF THE READ BUFFER. PROGRAM DEFAULT IS LOCATION 6700.
WRITE ADRESS	710	CONTAINS ADRESS OF THE WRITE BUFFER. THE PROGRAM CONTAINS 4 WRITE PATTERNS CONTIGIOUS IN CORE. LOCATION 2700 = ALL 1'S PATTERN LOCATION 3700 = ALTERNATE 1 AND 0 BYTES LOCATION 4700 = ALTERNATE 1 AND 0 BITS LOCATION 5700 = SLIDING 1 BIT PATTERN PROGRAM DEFAULT IS LOCATION 2700
ERASE COUNT	712	CONTAINS A 2'S COMPLIMENT NUMBER PROPORTIONAL TO THE AMO OF TAPE TO BE ERASED. THIS NUMBER IS LOADED INTO THE WORD COUNT REGISTER PRIOR TO AN ERASE COMMAND BIENG PERFORMED. PROGRAM DEFAULT IS 77777 . EACH INCREMENT CAUSES .02 INCHES OF TAPE TO BE ERASED.
OPERATION DELAY	714	CONTAINS A NUMBER USED IN A TIMER BETWEEN OPERATIONS DEFAULT =000001 MINIMUM DELAY
OPERATION DELAY MULT.	716	THIS IS USED IN CONJUNCTION WITH LOC. 714 AS A MULTIPLIER IN THE DELAY TIMER. DEFAULT IS 000001 MINIMUM DELAY. INCREASING THIS NUMBER WILL ALLOW MORE TIME BETWEEN OPERATIONS.

393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448

OPERATIONS NUMBER	720	THIS LOCATION CONTAINS THE NUMBER OF OPERATIONS TO BE PERFORMED IN THE OP TABLE. DEFAULT = 1 .
OPERATIONS TABLE	722 THRU 766	THIS IS THE BEGINNING OF THE OPERATIONS TABLE. ALL OPERATIONS TO BE PERFORMED SHOULD BE ENTERED IN THE DESIRED SEQUENCE IN THIS TABLE. DEFAULT IS A WRITE OPERATION.

7.2 ERROR CHECKS AND HALTS

LOCATION	DESCRIPTION
-----	-----
1320	HALT HERE IF THERE WAS AN ATTEMPT TO EXECUTE AN ILLEGAL FUNCTION, DUE TO TAPE POSITION OR SEQUENCE OF INSTRUCTIONS. THE ILLEGAL COMMAND IS DISPLAYED IN RO WHEN THE PROGRAM HALTS. SEE SECTION 5.1B FOR ILLEGAL FUNCTIONS
1332	HALT HERE IF THERE WAS A HARDWARE ERROR ON THE PREVIOUS OPERATION IF IT IS DESIRED TO BYPASS THE ERROR FLAG NOP THIS LOCATION. THE COMMAND AND STATUS REGISTER SHOULD BE EXAMINED AT THIS TIME TO DETERMINE THE PROBABLE CAUSE OF THE ERROR. PRESSING CONTINUE WILL CLEAR THE ERROR BY EXECUTING A CONTROL RESET.
1350	HALT HERE IF YOUR OPERATION TABLE LOC.722-766 HAS AN OPERATION THAT IS NOT DEFINED IN THE LEGAL FUNCTION CODES. RO HAS THE BAD CODE IN IT,CHECK YOUR TABLE IN LOCATIONS 722 THRU 766.
1406	HALT HERE IF BIT 15 OF THE SWR IS SET. THIS IS THE HALT BETWEEN INSTRUCTIONS.
1432	HALT HERE IF BIT 14 OF THE SWR IS SET. THIS IS THE HALT BETWEEN PASSES OF INSTRUCTIONS IN THE OP TABLE.

7.3 TABLE OF LEGAL FUNCTIONS AND CODES FOR USE IN OPERATIONS TABLE (LOC 722-766)

CODE	FUNCTION
----	-----
00	**** ILLEGAL ****
01	WRITE DATA (ILLEGAL IF EXECUTED FROM LOAD POINT)

449	02	READ (DATA, TAPE MARK OR I.D.B.)
450		
451	03	**** ILLEGAL ****
452		
453	04	SPACE REVERSE (ILLEGAL IF ISSUED FROM LOAD POINT)
454		
455	05	**** ILLEGAL ****
456		
457	06	**** ILLEGAL ****
458		
459	07	ERASE
460		
461	10	REWIND (TAPE MOVES AT 160 I.P.S.) ILLEGAL IF ISSUED FROM LOAD POINT.
462		
463	11	**** ILLEGAL ****
464		
465	12	**** ILLEGAL ****
466		
467	13	FAST FORWARD (TAPE MOVES FORWARD AT 160 I.P.S.)
468		
469	14	**** ILLEGAL ****
470		
471	15	WRITE I.D.B. (ILLEGAL IF ISSUED AT OTHER THAN LOAD POINT)
472		
473	16	**** ILLEGAL ****
474		
475	17	WRITE TAPE MARK (ILLEGAL IF ISSUED FROM LOAD POINT)
476		
477	20	CONTROL RESET (PROGRAM PSEUDO OP)

%
.ENABLE ABS

B. PROGRAM LISTING

.TITLE TR79 UTILITY DRIVER
.ASECT

* GENERAL REGISTER DEFINITIONS *

R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7

000000

000000
000001
000002
000003
000004
000005
000006
000007

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
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560

164000
164002
164004
164006

177776
177570

177564
177566

000000
000200

000200 000200
000204 000137 000230
000204 005237 000356

BEGIN:
BEGIN2:

* TR79 REGISTER DEFINITIONS *

TRCOM=164000
TRSTAT=164002
TRWC=164004
TRBA=164006

* PROCESSOR REGISTER DEFINITIONS *

PSW=177776
SWR=177570

* TTY REGISTERS *

TTSTAT=177564
TTBUF=177566

* TRAP CATCHERS *

. =0
. REPT 200
. +2
HALT
. ENDR

* STARTS AND CORE BUFFER BUILD *

. =200
JMP @#CORBIL ;NORMAL START
INC @#NORST ;SETS FLAG TO DETERMINE WHO STARTED

MO1

561 000210 000137 000230
562 000214 000777
563 000216 000777
564 000220 000400
565 000222 000777
566 000224 000525
567 000226 000652
568 000230 012701 000214
569 000234 012703 000216
570 000240 012702 002700
571 000244 012700 177400
572 000250 011122
573 000252 005200
574 000254 001403
575 000256 011322
576 000260 005200
577 000262 001372
578 000264 062701 000004
579 000270 062703 000004
580 000274 022701 000230
581
582
583
584 000300 001361
585 000302 005000
586 000304 005001
587 000306 005200
588 000310 005201
589 000312 010022
590 000314 022702 006700
591 000320 001405
592 000322 022701 000011
593 000326 001765
594 000330 006300
595 000332 000766
596 000334 005737 000356
597 000340 001404
598 000342 005037 000356
599 000346 000137 001460
600 000352 000137 001000
601 000356 000000
602
603
604
605
606
607 000500 000500
608 000500 000000
609
610
611
612 000600
613
614
615
616

JMP @CORBIL ;START HERE FOR DELAY ROUTINES
000777
000777
000400
000777
000525
000652
CORBIL: MOV #214,R1 ;SET UP PATTERN GENERATOR
MOV #216,R3 ;SET UP PATTERN GENERATOR
X1: MOV #2700,R2 ;SET UP ADDRESS POINTER
X2: MOV #177400,R0 ;SET UP COUNTER
X3: MOV (R1),(R2)+ ;DO IT
INC R0 ;KEEP TRACK OF HOW MANY
BEQ X4 ;CHECK FOR DONE
MOV (R3),(R2)+ ;DO IT
INC R0 ;KEEP COUNTING
BNE X3 ;LOOP HERE
X4: ADD #4,R1 ;NEXT PATTERN
ADD #4,R3 ;GO TO NEXT PATTERN
CMP #230,R1 ;DONE YET
;
;
BNE X2 ;NOT DONE YET
SLB: CLR R0 ;CLEAR THE PATTERN GENERATOR
CLR R1 ;CLEAR THE PATTERN COUNTER
INC R0 ;SET BIT IN PATTERN
SLB1: INC R1 ;KEEP COUNT
MOV R0,(R2)+ ;PUT IT IN CORE
CMP #6700,R2 ;SEE IF WERE FINISHED
BEQ SLBDON ;YES JUMP OUT
CMP #11,R1 ;CHECK ON THE BIT POSITION
BEQ SLB ;RESET THE SLIDING BIT
ASL R0 ;SHIFT THE BIT
BR SLB1 ;LOOP AGAIN
SLBDON: TST @#NORST ;SEE WHERE THE START CAME FROM
BEQ ALD ;IF = 0 MUST HAVE BEEN START 200
CLR @#NORST ;CLEAR IT OUTFOR NEXT TIME
JMP @#DRTN ;NOT = GO TO 204 START
JMP @#START ;GO TO A 200 START
ALD: ;TEMPORARY LOCATION
NORST: 000000

* STACKS *

SUBSTK: 000000 .=500 ;SUBROUTINE STACK
;
;
STACK: 000000 .=600 ;SET STACK HERE
;
;
;

* PROGRAM CONSTANTS AND VARIABLES *

617				
618				
619				
620				
621				
622				
623				
624		000664		.=664
625	000664	000000	SWRTEM:	000000
626	000666	000000	TIMMUL:	000000
627	000670	000000	COMTEM:	000000
628	000672	000000	STTEM:	000000
629	000674	000000	TEMP1:	000000
630	000676	000000	TEMP2:	000000
631	000700	000000	EXTCOR:	000000
632	000702	000000	UNIT:	000000
633	000704	177760	WCNT:	-20
634	000706	006700	RADDR:	006700
635	000710	002700	WADDR:	002700
636	000712	177777	ERSCNT:	177777
637	000714	000001	OPDLY:	000001
638	000716	000001	OPDLX:	000001
639	000720	000001	OPNUM:	000001
640				
641				
642				
643				
644				
645	000722	000001	OPTBL:	1
646	000724	000000		0
647	000726	000000		0
648	000730	000000		0
649	000732	000000		0
650	000734	000000		0
651	000736	000000		0
652	000740	000000		0
653	000742	000000		0
654	000744	000000		0
655	000746	000000		0
656	000750	000000		0
657	000752	000000		0
658	000754	000000		0
659	000756	000000		0
660	000760	000000		0
661	000762	000000		0
662	000764	000000		0
663	000766	000000		0
664				
665				
666				
667				
668				
669				
670				
671				
672				

```

:TEMP STORAGE FOR SWITCH REGISTER
:CONTAINS DELAY MULTIPLIERS
:TEMP STORAGE FOR COMMAND REGISTER
:TEMP STORAGE FOR STATUS REGISTER
:TEMP STORAGE FOR THE NUMBER OF OPERATIONS TO BE PERFORMED
:TEMP STORAGE FOR THE OPERATION BIENG PERFORMED
:BITS 12 AND 13 ARE XBA 16AND XBA 17
:UNIT SELECT BITS 8 AND 9
:WORD COUNT
:START OF READ BUFFER
:START OF WRITE BUFFER
:CONTAINS LENGTH OF ERASURE WHEN DOING AN ERASE
:OPERATION DELAY
:OPERATION DELAY MULTIPLIER
:NUMBER OF OPERATIONS TO BE PERFORMED

```

* TABLE OF OPERATIONS TO BE PERFORMED MAX =23 *

```

:1ST OPERATION
:2ND OPERATION
:3RD OPERATION
:4TH OPERATION
:5TH OPERATION
:6TH OPERATION
:7TH OPERATION
:10TH OPERATION
:11TH OPERATION
:12TH OPERATION
:13TH OPERATION
:14TH OPERATION
:15TH OPERATION
:16TH OPERATION
:17TH OPERATION
:20TH OPERATION
:21ST OPERATION
:22ND OPERATION
:23RD OPERATION

```

* HOUSE KEEPING AND INITIAL PROGRAM PARAMETERS *

```

673      001000 001000
674 001000 012706 000600      START:  MOV #600,SP      ;SET UP STACK AT LOC. 600
675 001004 012737 000340 177776  MOV #340,@#PSW      ;SET PRIORITY LEVEL 7 NO INTERRUPTS ALLOWED
676 001012 032737 004000 164000 1$: BIT #4000,@#TRCOM ;WAIT FOR THE INITIAL 900 MS.POWER CLEAR TO SUBSIDE
677 001020 001374      BNE 1$      ;LOOP UNTILL CONTROL IS READY
678 001022 013701 000720      MOV @#OPNUM,R1      ;R1 CONTAINS HOW MANY OPERATIONS WILL BE DONE
679 001026 042737 176377 000702 UNUM: BIC #176377,@#UNIT ;STRIP OFF EVERYTHING EXCEPT BITS 8 AND 9 TO SET UP UNI
680
681
682
683
684
685
686 001034 012702 000722      MOV #OPTBL,R2      ;R2 CONTAINS THE POINTER TO THE START OF THE OPERATINS
687 001040 012737 000011 000674 LOOP: MOV #9,@#TEMP1      ;THE NUMBER OF LEAGAL OPERATIONS
688 001046 012703 001440      MOV #LEGOPS,R3      ;POINTER TO THE BEGINNING OF THE LEGAL OPERATIONS COMPA
689 001052 012237 000676      MOV (R2)+,@#TEMP2 ;GET THE OPERATION AND PUT IT IN TEMP 2
690 001056 023723 000676 2$: CMP @#TEMP2,(R3)+ ;CHECK IT AGAINST THE LEGAL FUNCTIONS
691 001062 001422      BEQ CLINH      ;IF ITS LEGAL CONTINUE TO EXECUTE IT
692 001064 005367 177604      DEC TEMP1      ;WASN'T THAT OPERATION SUBTRACT 1
693 001070 001372      BNE 2$      ;TRY NEXT ONE
694
695
696
697
698
699
700
701
702 001072 022737 000020 000676      CMP #20,@#TEMP2 ;SEE IF ITS A CONTROL RESET (PSEUDO OP)
703 001100 001402      BEQ CRES      ;MUST BE A CONTROL RESET JUMP TO CRES
704 001102 000137 001344      JMP @#ILFUT      ;IT MUST BE ILLEGAL FUNCTION STOP THE PROGRAM
705 001106 052767 004000 162664 CRES: BIS #4000,TRCOM ;MUST BE A POWER CLEAR SO DO IT TAKES 900 MILI SECONDS
706 001114 032767 004000 162656 3$: BIT #4000,TRCOM ;SEE IF DONE WITH POWER CLEAR YET
707 001122 001374      BNE 3$      ;WAIT UNTILL DONE
708 001124 000167 000134      JMP FUDONE      ;GET BACK INTO PROGRAM
709
710
711
712
713
714
715
716 001130 005037 164002      CLINH: CLR @#TRSTAT ;CLEAR OUT THE INHIBIT BIT
717 001134 162703 001442      SUB #LEGOPS+2,R3 ;GET AN OFFSET VALUE
718 001140 060307      ADD R3,PC ;ADD IT TO THE PC AND GO THERE
719
720
721
722
723
724
725
726 001142 000422      W: BR WRITE ;WRITE INSTRUCTION
727 001144 000425      R: BR READ ;READ INSTRUCTION
728 001146 000401      SR: BR .+4 ;SPACE REVERSE INSTRUCTION

```

* SET UP OPERATIONS COMPARITOR *

* CHECK FOR A CONTROL RESET *

* CLEAR INHIBIT BIT AND SET UP OPERATION *

* OPERATIONS DIRECTORY TABLE *

729 001150 000432
730 001152 000240
731 001154 000240
732 001156 000240
733 001160 000240
734
735
736
737
738
739
740
741

ER: BR ERSE ;ERASE COMMAND
REWD: NOP ;REWIND COMMAND
FFOR: NOP ;FAST FORWARD COMMAND
IDB: NOP ;WRITE ID BURST
WTM: NOP ;WRITE TAPE MARK

* SET UP UNIT NUMBER AND GO BIT *

742 001162 006337 000676
743 001166 053737 000702 000676
744 001174 053737 000700 000676
745 001202 005237 000676
746 001206 000417
747
748
749
750
751
752
753
754
755

SGOB: ASL @#TEMP2 ;SHIFT THE FUNCTION INTO THE PROPER BIT POSITIONS
BIS @#UNIT,@#TEMP2 ;SET THE UNIT # BITS
BIS @#EXTCOR,@#TEMP2 ;SET THE MEMORY EXTENSION BITS
INC @#TEMP2 ;SET THE GO BIT
BR EXECUT ;JUMP TO THE EXECUTION ROUTINE

* OPERATIONS SETUP ROUTINES *

756 001210 013737 000710 164006
757 001216 000403
758 001220 013737 000706 164006
759 001226 013737 000704 164004
760 001234 000752
761 001236 013737 000712 164004
762 001244 000746
763
764
765
766
767
768
769

WRITE: MOV @#WADDR,@#TRBA ;SET UP WRITE BUFFER AREA
BR WRIWC ;CONTINUE ON TO SET UP W.C.
READ: MOV @#RADDR,@#TRBA ;SET UP READ BUFFER ADDRESS
WRIWC: MOV @#WCNT,@#TRWC ;SET UP WORD COUNT
BR SGOB ;SET UP GO BIT
ERSE: MOV @#ERSCNT,@#TRWC ;SET NUMBER TO INDICATE AMOUNT OF TAPE TO BE ERASED
BR SGOB ;SET UP GO BIT

* ROUTINE TO EXECUTE THE FUNCTION AND CHECK FOR DONE *

770 001246 000240
771 001250 013737 000676 164000
772 001256 105737 164000
773 001262 100375
774
775
776
777
778
779
780
781
782

EXECUT: NOP ;DO THE FUNCTION
MOV @#TEMP2,@#TRCOM ;SEE IF DONE YET
7\$: TSTB @#TRCOM ;WAIT FOR IT
BPL 7\$

* ROUTINE TO CHECK FUNCTION WHEN DONE *

783 001264 013737 164002 000672
784 001272 013737 164000 000670

FUDONE: MOV @#TRSTAT,@#STTEM ;SAVE STATUS
MOV @#TRCOM,@#COMTEM ;SAVE COMMAND REGISTER


```

785 001300 032737 040000 000670 BIT #40000, @#COMTEM ; WAS IT AN ILLEGAL COMMAND DUE TO SEQUENCE OR TAPE POSI
786 001306 001405 BEQ ERDONE ; NO ERROR HERE
787 001310 006237 000676 ILLCOM: ASR @#TEMP2 ; STRIP OFF THE GO BIT
788 001314 013700 000676 MOV @#TEMP2, R0 ; PUT BAD COMMAND IN R0
789 001320 000000 ERR14: HALT ; STOP WITH BAD COMMAND DISPLAYED
790 001322 032737 100000 000670 ERDONE: BIT #100000, @#COMTEM ; SEE IF ERROR BIT IS SET
791 001330 001410 BEQ OPDEL ; NO ERRORS CONTINUE
792 001332 000000 ERR15: HALT ; GOT AN ERROR NOP THIS HALT TO CONTINUE
793 001334 000137 001106 JMP @#CRES ; IF YOU GOT AN ERROR ONLY RECOVERY IS WITH A CONTROL RE
794 001340 006237 000676 ILLFUN: ASR @#TEMP2 ; STRIP OFF GO BIT
795 001344 013700 000676 ILFUT: MOV @#TEMP2, R0 ; PUT THE BAD CODE IN R0 TO DISPLAY WHEN HALTED
796 001350 000000 ERRIF: HALT ; GOT AN ILLEGAL FUNCTION CHECK YOUR PROGRAM LOCATION 72
797
798
799
800
801
802
803
804
805

```

* OPERATION DELAY BETWEEN INSTRUCTIONS *

```

806 001352 013737 000716 000666 OPDEL: MOV @#OPDLX, @#TIMMUL ; SET OP OPERATIONS DELAY MULTIPLIER
807 001360 013700 000714 MOV @#OPDLY, R0 ; SET UP OPERATIONS DELAY TIMER
808 001364 005300 8$: DEC R0 ; TIMER IS TICKING
809 001366 001376 BNE 8$ ; GET MORE TIME
810 001370 005337 000666 DEC @#TIMMUL ; COUNT DOWN THE MULTIPLIER
811 001374 001373 BNE 8$ ; GET MORE TIME
812 001376 032737 100000 177570 BIT #100000, @#SWR ; TIMES UP CHECK SWITCHES TO SEE IF WE HALT OR CONTINUE
813 001404 001401 BEQ .+4 ; DONT STOP NOW SKIP THE HALT
814 001406 000000 INSHLT: HALT ; STOP BETWEEN INSTRUCTIONS
815 001410 005301 DEC R1 ; -1 FROM THE NUMBER OF OPERATIONS IN R1
816 001412 001001 BNE 9$ ; GO AND DO THE NEXT INSTRUCTION
817 001414 001040 BR .+4 ; SKIP THE JUMP
818 001416 0010137 001040 9$: JMP @#LOOP ; DO THE LOOP AGAIN
819 001422 032737 040000 177570 BIT #40000, @#SWR ; CHECK SWITCHES TO SEE IF WE WANT TO STOP AT END OF PAS
820 001430 0011401 BEQ REST ; DO THE NEXT PASS SKIP THE HALT
821 001432 000000 PASHLT: HALT ; STOP BETWEEN PASSES
822 001434 0010137 001000 REST: JMP @#START ; GO DO IT AGAIN (NEXT PASS)
823
824
825
826
827
828

```

* LEGAL OPERATIONS COMPARITOR TABLE *

```

829 001440 000001 LEGOPS: 00001 ; WRITE
830 001442 000002 00002 ; READ
831 001444 000004 00004 ; SPACE REVERSE
832 001446 000007 00007 ; ERASE
833 001450 000010 00010 ; REWIND
834 001452 000013 00013 ; FAST FORWARD
835 001454 000015 00015 ; WRITE IDB
836 001456 000017 00017 ; WRITE TAPE MARK
837
838
839
840

```

841
842
843
844
845
846 001460
847 001460 012706 000600
848 001464 012705 000500
849 001470 012737 000340 177776
850 001476 032737 004000 164000
851 001504 001774
852 001506 013737 177570 000664
853 001514 105737 000664
854 001520 100372
855 001522 042737 177760 000664
856 001530 006337 000664
857 001534 006337 000664
858 001540 063707 000664
859
860
861
862
863
864
865
866
867
868

```

*****
*   DELAY SET-UP PROGRAM ROUTINES   *
*****

      =1460
DRTN:  MOV #600,SP           ;SET STACK
      MOV #500,R5          ;SET UP SUBROUTINE STACK
      MOV #340,@#PSW       ;SET PRIORITY 7 NO INTERRUPTS ALLOWED
DROUTS: BIT #4000,@#TRCOM  ;CHECK FOR INITIAL POWER CLEAR TO SUBSIDE
      BEQ DROUTS          ;WAIT TILL DONE
BR1:   MOV @#SWR,@#SWRTEM  ;GET SWITCHES AND PUT THEM IN STORAGE
      TSTB @#SWRTEM       ;SEE IF ENABLE IS UP YET
      BPL BR1             ;LOOP UNTILL ENABLE IS UP
      BIC #177760,@#SWRTEM ;MASK BITS AND GET A NUMBER BETWEEN 0-16
      ASL @#SWRTEM        ;SHIFT IT LEFT TO MULTIPLY BY 2
      ASL @#SWRTEM        ;SHIFT AGAIN MULTIPLY BY 2 TO GET OFFSET
      ADD @#SWRTEM,PC     ;ADD IT TO THE PC AND GO THERE

```

869 001544 000137 001506
870 001550 000137 001644
871 001554 000137 001644
872 001560 000137 001662
873 001564 000137 001722
874 001570 000137 001764
875 001574 000137 001764
876 001600 000137 001764
877 001604 000137 002120
878 001610 000137 002174
879 001614 000137 002174
880 001620 000137 002314
881 001624 000137 002040
882 001630 000137 001722
883 001634 000137 002234
884 001640 000137 001506
885
886
887
888
889
890
891
892
893
894

```

*****
*   INDEX TABLE OF DELAY PROGRAM DIRECTIVES   *
*****

TABLE:  JMP @#BR1           ;DELAY 0 IS A NO-OP
        JMP @#BR2         ;DELAY 1 IS FOR POWER CLEAR
        JMP @#BR2         ;DELAY 2 IS FOR POWER CLEAR
        JMP @#BR3         ;DELAY 3 IS FOR ERROR CLK
        JMP @#BR4         ;DELAY 4 IS FOR WRITE ENABLE
        JMP @#BR5         ;DELAY 5 IS FOR OUT BUFF FLAG
        JMP @#BR5         ;DELAY 6 IS FOR END WRITE DATA
        JMP @#BR5         ;DELAY 7 IS FOR FIRST WORD WRITE REQUEST
        JMP @#BR10        ;DELAY 10 IS FOR ERASE
        JMP @#BR11        ;DELAY 11 IS FOR WRITE I.D.B.
        JMP @#BR11        ;DELAY 12 IS FOR I.D.B. TIMING
        JMP @#BR16        ;DELAY 13 IS FOR ABORT WINDOW
        JMP @#BR6         ;DELAY 14 IS FOR LD CTRL + BUSY
        JMP @#BR4         ;DELAY 15 IS FOR GO PULSE DELAY
        JMP @#BR14        ;DELAY 16 IS FOR M.S.D. DELAY
        JMP @#BR1         ;DELAY 17 IS A NO-OP

```

895 001644 004537 002442
896 001650 105737 177570

```

*****
*   ROUTINE FOR DELAYS 1 AND 2   *
*****

BR2:   JSR R5,@#PCL       ;DO A POWER CLEAR
      TSTB @#SWR         ;CHECK FOR A LOOP

```

```

897 001654 100773          BMI BR2          ;DO IT AGAIN
898 001656 000137 001506  JMP @#BR1        ;GET NEXT TEST
899
900
901
902
903
904 001662 012700 000070  BR3:  MOV #70,R0      ;SET UP DELAY MULTIPLIER
905 001666 052737 004000 164002  BIS #4000,@#TRSTAT ;FORCE AN ERROR WITH B.G.L. BIT
906 001674 005300          13$:  DEC R0          ;TIMER IS TICKING
907 001676 001376          BNE 13$        ;CHECK TIMER
908 001700 005037 164002  CLR @#TRSTAT   ;OK NOW CLEAR THE BIT
909 001704 105737 177570  TSTB @#SWR    ;SEE IF WE WANT TO DO IT AGAIN
910 001710 100764          BMI BR3        ;OK LOOP BACK
911 001712 004537 002442  JSR R5,@#PCL  ;DONE HERE DO A POWER CLEAR AND GET THE NEXT ONE
912 001716 000137 001506  JMP @#BR1     ;GET THE NEXT DELAY DIRECTIVE
913
914
915
916
917
918 001722 004537 002462  BR4:  JSR R5,@#RDY  ;CHECK FOR READY
919 001726 012737 177777 164004  BR4A: MOV #-1,@#TRWC ;THIS NUMBER IS USED FOR AN ERASE COUNT
920 001734 012737 000017 164000  MOV #17,@#TRCOM ;DO THE ERASE
921 001742 004537 002462  JSR R5,@#RDY  ;WAIT TILL DONE
922 001746 004537 002500  JSR R5,@#ERCK ;SEE IF WE ERRORED OUT
923 001752 105737 177570  TSTB @#SWR   ;SEE IF WE LOOPON TEST
924 001756 100763          BMI BR4A      ;LOOP HERE AND DO IT AGAIN
925 001760 000137 001506  JMP @#BR1     ;GET OUT AND GET NEXT DELAY DIRECTIVE
926
927
928
929
930
931
932 001764 004537 002530  BR5:  JSR R5,@#OFLP ;SEE IF WE ARE AT LOAD POINT
933 001770 005037 164002  CLR @#TRSTAT  ;CLEAR INHIBIT BIT
934 001774 012737 177774 164004  BR5A: MOV #-4,@#TRWC ;SET UP FOR A 10 BYTE WRITE
935 002002 013737 000710 164006  MOV @#WADDR,@#TRBA ;SET UP THE WRITE ADRESS BUFFER
936 002010 012737 000003 164000  MOV #3,@#TRCOM ;DO THE WRITE
937 002016 004537 002462  JSR R5,@#RDY  ;WAIT FOR READY
938 002022 004537 002500  JSR R5,@#ERCK ;CHECK FOR ERRORS
939 002026 105737 177570  TSTB @#SWR   ;SEE IF WE WANT TO LOOP
940 002032 100760          BMI BR5A      ;LOOP HERE AND DO IT AGAIN
941 002034 000137 001506  JMP @#BR1     ;GET OUT AND GET NEXT DELAY DIRECTIVE
942
943
944
945
946
947 002040 004537 002564  BR6:  JSR R5,@#REW  ;DO A REWIND GET TO B.O.T.
948 002044 004537 002614  BR6C: JSR R5,@#ETS  ;CHECK FOR END OF TAPE ANYWAY
949 002050 005037 164002  CLR @#TRSTAT  ;CLEAR THE INHIBIT BIT
950 002054 005037 164004  CLR @#TRWC    ;CLEAR THE WORD COUNT
951 002060 012737 000017 164000  MOV #17,@#TRCOM ;DO AN ERASE
952 002066 005037 164000  BR6A: CLR @#TRCOM ;FORCE A LOAD PULSE

```

```

953 002072 105737 177570      TSTB  @#SWR      ;SEE IF WE WANT TO LOOP
954 002076 100004      BPL   BR6B      ;ALL DONE
955 002100 105737 164000      TSTB  @#TRCOM   ;SEE IF ERASE IS DONE YET
956 002104 100370      BPL   BR6A      ;NOT DONE DO ANOTHER LOAD PULSE
957 002106 000756      BR    BR6C      ;DO IT AGAIN
958 002110 004537 002442      BR6B: JSR   R5,@#PCL ;DO A CLEAR AND EXIT
959 002114 000137 001506      JMP   @#BR1     ;GO BACK TO MAIN
960
961
962
963
964
965
966
967
968 002120 004537 002614      BR10: JSR   R5,@#ETS  ;CHECK FOR E.O.T.
969 002124 004537 002530      JSR   R5,@#OFLP ;GET US OFF LOAD POINT
970 002130 005037 164002      CLR   @#TRSTAT  ;CLEAR INHIBIT
971 002134 012737 000017      MOV   #17,@#TRCOM ;DO A MAXIMUM ERASE
972 002142 012700 005000      BR10A: MOV  #5000,R0 ;SET UP COUNTER
973 002146 005300      BR10B: DEC   R0      ;START COUNTDOWN
974 002150 001376      BNE   BR10B     ;TIMER IS TICKING
975 002152 005037 164004      CLR   @#TRWC    ;RE ESTABLISH THE ERASE COUNT TO 0
976 002156 105737 177570      TSTB  @#SWR     ;SEE IF WE LOOP HERE
977 002162 100767      BMI   BR10A     ;DO IT AGAIN
978 002164 004537 002442      JSR   R5,@#PCL  ;DO A POWER CLEAR
979 002170 000137 001506      JMP   @#BR1     ;GET OUT AND GET THE NEXT DELAY DIRECTIVE
980
981
982
983
984
985
986
987
988
989
990
991 002174 004537 002564      BR11: JSR   R5,@#REW  ;DO A REWIND
992 002200 012737 000033      MOV   #33,@#TRCOM ;WRITE AN I.D.B.
993 002206 004537 002462      JSR   R5,@#RDY  ;CHECK FOR DONE
994 002212 004537 002500      JSR   R5,@#ERCK ;SEE IF ANY ERRORS UP
995 002216 105737 177570      TSTB  @#SWR     ;SEE IF WE WANT TO LOOP
996 002222 100764      BMI   BR11     ;LOOP BACK DO IT AGAIN
997 002224 004537 002442      JSR   R5,@#PCL  ;DO A POWER CLEAR
998 002230 000137 001506      JMP   @#BR1     ;GET OUT AND GET NEXT DELAY DIRECTIVE
999
1000
1001
1002
1003
1004
1005
1006
1007
1008 002234 105737 164002      BR14: TSTB  @#TRSTAT ;CHECK FOR EOT UP

```

* ROUTINE FOR DELAY 10 *

* ROUTINE FOR DELAY 11 AND 12 *

* ROUTINE FOR DELAY 16 *

```

1009 002240 100407          BMI BR14A          ;SKIP THE FAST FORWARD
1010 002242 005037 164002  CLR @#TRSTAT      ;CLEAR INHIBIT
1011 002246 012737 000027 164000  MOV #27,@#TRCOM   ;DO A FAST FORWARD
1012 002254 004537 002462      JSR R5,@#RDY      ;WAIT TILL DONE
1013 002260 005037 164002      BR14A: CLR @#TRSTAT ;CLEAR INHIBIT
1014 002264 012737 000027 164000  MOV #27,@#TRCOM ;TRY A FAST FORWARD ,SHOULD PRODUCE ERROR
1015 002272 004537 002462      JSR R5,@#RDY      ;WAIT TILL DONE
1016 002276 105737 177570      TSTB @#SWR        ;SEE IF WE LOOP HERE
1017 002302 100766          BMI BR14A          ;YES LOOP HERE
1018 002304 004537 002442      JSR R5,@#PCL      ;DO A POWER CLEAR
1019 002310 000137 001506      JMP @#BR1         ;GET OUT DO NEXT DELAY

```

```

*****
* ROUTINE FOR DELAY 13 *
*****

```

```

1020 ;
1021 ;
1022 ;
1023 ;
1024 ;
1025 ;
1026 ;
1027 002314 004537 002564      BR16: JSR R5,@#REW   ;DO A REWIND
1028 002320 005037 164002      CLR @#TRSTAT      ;CLEAR INHIBIT
1029 002324 012737 177000 164004  MOV #177000,@#TRWC ;SET UP ERASE COUNT
1030 002332 012737 000017 164000  MOV #17,@#TRCOM   ;DO AN ERASE
1031 002340 004537 002462      JSR R5,@#RDY      ;WAIT FOR IT
1032 002344 004537 002564      JSR R5,@#REW      ;REWIND IT
1033 002350 004537 002530      JSR R5,@#OFLP     ;GET OFF LOAD POINT LEGALLY
1034 002354 012737 177760 164004  BR16A: MOV #-20,@#TRWC ;SET UP W.C.
1035 002362 012737 003600 164006  MOV #3600,@#TRBA  ;SET UP CORE ADDRESS
1036 002370 005037 164002      CLR @#TRSTAT      ;CLEAR THE INHIBIT
1037 002374 012737 000003 164000  MOV #3,@#TRCOM    ;NOW TRY TO WRITE NOTHING,SHOULD ABORT
1038 002402 012701 000010      BR16B: MOV #10,R1   ;SET UP TIME MULTIPLIER
1039 002406 005000          BR16D: CLR R0      ;CLEAR TIMER
1040 002410 005200          BR16C: INC R0        ;TIMES WAITING
1041 002412 001376          BNE BR16C         ;TIMER RUNNING
1042 002414 005301          DEC R1            ;-1 FROM MULTIPLIER
1043 002416 001373          BNE BR16D        ;SEE IF DONE YET
1044 002420 004537 002442      JSR R5,@#PCL      ;CLEAR ERRORS
1045 002424 105737 177570      TSTB @#SWR        ; LOOP ???
1046 002430 100751          BMI BR16A         ;DO IT AGAIN
1047 002432 004537 002442      JSR R5,@#PCL      ;POWER CLEAR
1048 002436 000137 001506      JMP @#BR1         ;ALL DONE GET SOME MORE

```

```

*****
* SUBROUTINE FOR POWER CLEAR *
*****

```

```

1049 ;
1050 ;
1051 ;
1052 ;
1053 ;
1054 ;
1055 ;
1056 ;
1057 ;
1058 ;
1059 002442 052737 004000 164000  PCL: BIS #4000,@#TRCOM ;SET POWER CLEAR
1060 002450 032737 004000 164000  PCL1: BIT #4000,@#TRCOM ;WAIT FOR 900 MILI SECONDS
1061 002456 001374          BNE PCL1          ;STILL WAITING
1062 002460 000205          RTS R5            ;RETURN TO MAIN ROUTINE
1063 ;
1064 ;

```

```

1065
1066
1067
1068
1069
1070
1071 002462 105737 164000 RDY: TSTB @#TRCOM ;CHECK ON DONE BIT
1072 002466 100375 BPL RDY ;WAIT TILL DONE
1073 002470 042737 000001 164002 IHB: BIC #1,@#TRSTAT ;CLEAR THE INHIBIT BIT
1074 002476 000205 RTS R5 ;RETUN TO MAIN ROUTINE
1075
1076
1077
1078
1079
1080
1081
1082
1083 002500 005737 164000 ERCK: TST @#TRCOM ;SEE IF ERROR IS UP
1084 002504 100010 BPL ERCK2 ;NO ERRORS JUMP OUT
1085 002506 012737 000007 177566 ERCK1: MOV #7,@#TTBUF ;GOT AN ERROR RING A BELL
1086 002514 105737 177564 TSTB @#TTSTAT ;WAIT HERE
1087 002520 100375 BPL ERCK1 ;WAIT HERE
1088 002522 004537 002442 JSR R5,@#PCL ;CLEAR THE ERROR WITH A POWER CLEAR
1089 002526 000205 RTS R5 ;GO BACK TO MAIN
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099 002530 032737 000040 164002 OFLP: BIT #40,@#TRSTAT ;SEE IF LOAD POINT IS UP
1100 002536 001411 BEQ OFLP1 ;NO LP JUMP OUT
1101 002540 005037 164002 CLR @#TRSTAT ;CLEAR THE INHIBIT
1102 002544 012737 000033 164000 MOV #33,@#TRCOM ;WRITE AN I.D.B.
1103 002552 004537 002462 JSR R5,@#RDY ;WAIT FOR READY
1104 002556 004567 177716 JSR R5,ERCK ;CHECK FOR ERRORS
1105 002562 000205 OFLP1: RTS R5 ;GO BACK TO MAIN
1106
1107
1108
1109
1110
1111
1112
1113 002564 032737 000040 164002 REW: BIT #40,@#TRSTAT ;AT LOAD POINT??
1114 002572 001007 BNE REW2 ;YES DON'T NEED REWIND
1115 002574 005037 164002 REW1: CLR @#TRSTAT ;CLR THE INHIBIT
1116 002600 012737 000021 164000 MOV #21,@#TRCOM ;DO A REWIND
1117 002606 004537 002462 JSR R5,@#RDY ;WAIT TILL DONE
1118 002612 000205 REW2: RTS R5 ;GO BACK
1119
1120

```

* SUBROUTINE FOR READY AND CLEAR INHIBIT *

* SUBROUTINE TO CHECK FOR ERRORS *

* SUBROUTINE TO GET OFF LOAD POINT LEGALLY *

* SUBROUTINE FOR REWIND AND L.P. *

* SUBROUTINE FOR E.O.T. *

1121
1122
1123
1124 002614 105737 164002
1125 002620 100002
1126 002622 004537 002564
1127 002626 000205

ETS: TSTB @#TRSTAT ; IS END OF TAPE UP ?
BPL ETS1 ; NOT AT E.O.T.
JSR R5,@#REW ; DO A REWIND
ETS1: RTS R5 ; GET BACK

* WRITE BUFFER PATTERNS *

. =2700
. REPT 1000 ; ALL 1'S
. WORD 000777 ; PATTERN = 0 000 000 111 111 111
. ENDR

. =3700
. REPT 1000 ; 1'S AND 0'S ALTERNATE WORDS
. WORD 000400 ; PATTERN = 0 000 000 100 000 000
. WORD 000777 ; PATTERN = 0 000 000 111 111 111
. ENDR

. =4700
. REPT 1000 ; ALTERNATE BITS
. WORD 000525 ; PATTERN = 0 000 000 101 010 101
. WORD 000652 ; PATTERN = 0 000 000 110 101 010
. ENDR

. =5700
. REPT 1000 ; SLIDING 1 BIT
. WORD 000001 ; PATTERN = 0 000 000 000 000 001
. WORD 000002 ; PATTERN = 0 000 000 000 000 010
. WORD 000004 ; PATTERN = 0 000 000 000 000 100
. WORD 000010 ; PATTERN = 0 000 000 000 001 000
. WORD 000020 ; PATTERN = 0 000 000 000 010 000
. WORD 000040 ; PATTERN = 0 000 000 000 100 000
. WORD 000100 ; PATTERN = 0 000 000 001 000 000
. WORD 000200 ; PATTERN = 0 000 000 010 000 000
. WORD 000400 ; PATTERN = 0 000 000 100 000 000
. ENDR

* READ BUFFER AREA *

1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176

K02

TR79 UTILITY DRIVER MACY11 27(1006) 22-DEC-76 09:23 PAGE 1-21
DZTRBS.M11 22-DEC-76 09:06

SEQ 0022

1177
1178
1179
1180
1181

000001

⋮

.=6700

;1000 WORD LOCATIONS RESERVED FOR READ BUFFER

.END

ALD	000352	BR6A	002066	EXTCOR	000700	PSW	= 177776	TEMP2	000676
BEGIN	000200	BR6B	002110	FFOR	001154	R	001144	TIMMUL	000666
BEGIN2	000204	BR6C	002044	FUDONE	001264	RADDR	000706	TRBA	= 164006
BR1	001506	CLINH	001130	IDB	001156	RDY	002462	TRCOM	= 164000
BR10	002120	COMTEM	000670	IHB	002470	READ	001220	TRSTAT	= 164002
BR10A	002142	CORBIL	000230	ILFUT	001344	REST	001434	TRWC	= 164004
BR10B	002146	CRES	001106	ILLCOM	001310	REW	002564	TTBUF	= 177566
BR11	002174	DROUTS	001476	ILLFUN	001340	REWD	001152	TTSTAT	= 177564
BR14	002234	DRTN	001460	INSHLT	001406	REW1	002574	UNIT	000702
BR14A	002260	ER	001150	LEGOPS	001440	REW2	002612	UNUM	001026
BR16	002314	ERCK	002500	LOOP	001040	SGOB	001162	W	001142
BR16A	002354	ERCK1	002514	NORST	000356	SLB	000302	WADDR	000710
BR16B	002402	ERCK2	002526	OFLP	002530	SLBDON	000334	WCNT	000704
BR16C	002410	ERDONE	001322	OFLP1	002562	SLB1	000310	WRITE	001210
BR16D	002406	ERRIF	001350	OPDEL	001352	SR	001146	WR1WC	001226
BR2	001644	ERR14	001320	OPDLX	000716	START	001000	WTM	001160
BR3	001662	ERR15	001332	OPDLY	000714	STTEM	000672	X1	000240
BR4	001722	ERSCNT	000712	OPNUM	000720	SUBSTK	000500	X2	000244
BR4A	001726	ERSE	001236	OPTBL	000722	SWR	= 177570	X3	000250
BR5	001764	ETS	002614	PASHLT	001432	SWRTEM	000664	X4	000264
BR5A	001774	ETS1	002626	PCL	002442	TABLE	001544	.	= 002630
BR6	002040	EXECUT	001246	PCL1	002450	TEMP1	000674		

. ABS. 002630 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZTRBB, DZTRBB=DZTRBB.M11
RUN-TIME: 49.3 SECONDS
RUN-TIME RATIO: 53/14=3.6
CORE USED: 5K (9 PAGES)

M02

