

# TM03/TE16

DRIVE FUNCTION TIMER  
MD-11-DZTEE-A

EP-DZTEE-A-DL-A  
COPYRIGHT © 1977  
FICHE 1 OF 1

JUN 1977  
**digital**  
MADE IN USA

This microfiche card contains 48 frames of technical data, arranged in a 6x8 grid. The frames include various diagrams, tables, and text blocks related to the drive function timer. The data is too small to be legible in this image.

801

EOF1DZTCB0SEQ

00010000

770629

PDP10 411

BDH0R1DZTEEASE0

00010000

770629

CO1

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 1  
DZTEEA.P11 31-MAR-77 00:00

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTEE-A-D  
PRODUCT NAME: TMO3/TE16 DRIVE FUNCTION TIMER  
DATE CREATED: 15 FEB 77  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: J. G. ADAMS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS 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) 1977 BY DIGITAL EQUIPMENT CORPORATION

TH03 DRIVE FUNCTION TIMER  
TABLE OF CONTENTS

PAGE 2

TABLE OF CONTENTS

ABSTRACT

CHAPTER 1 REQUIREMENTS

- 1.1 EQUIPMENT
- 1.2 MEMORY STORAGE
- 1.3 PRELIMINARY PROGRAMS

CHAPTER 2 LOADING AND STARTING PROCEDURE

- 2.1 ACT11 OPERATION

CHAPTER 3 SWITCH SETTINGS

CHAPTER 4 ERRORS

- 4.1 ERROR TYPEOUT FORMAT (HARDWARE)
- 4.2 ERROR TYPEOUT FORMAT (FUNCTION OUT OF RANGE)

CHAPTER 5 SUBROUTINE ABSTRACTS

CHAPTER 6 MISCELLANEOUS

- 6.1 STACK POINTER
- 6.2 EXECUTION TIME

CHAPTER 7 PROGRAM DESCRIPTION

- 7.1 FUNCTION TIME DOCUMENT
- 7.2 TEST SEQUENCE / RELATED ADJUSTMENTS / ASSOCIATED HARDWARE
- 7.3 SUBTEST DESCRIPTIONS

EO1

DZTEE-A TM03/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 3  
DZTEEA.P11 31-MAR-77 00:00

TM03 DRIVE FUNCTION TIMER  
ABSTRACT

PAGE 3

ABSTRACT

PROGRAM DZTEE MEASURES THE TIME REQUIRED AND GAP SIZES PRODUCED BY THE TM03/TE16 MAGTAPE DRIVE/SLAVE.

THE TEST WILL CHECK BOTH THE LOGIC GENERATED TIME DELAYS, AND THE DISTANCES TRAVELED BY THE TAPE.

ACTUAL TAPE SPEED MAY ALSO BE CHECKED BY USING THE SPEED TESTS WITH AN 800 BPI SKEW TAPE.

DEVICE ERRORS ARE CHECKED AND PRINTED AS THEY OCCUR. IF AN ERROR IS DATA RELATED(PARITY; ETC) THEY ARE PRINTED AS SOFT ERRORS.

IF A TIME CHECK IS OUT OF RANGE, IT IS PRINTED AS AN OUT OF RANGE ERROR.

F01

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY:1 30(1046) 27-JUN-77 17:03 PAGE 4  
DZTEEA.P11 31-MAR-77 00:00

TMO3 DRIVE FUNCTION TIMER  
REQUIREMENTS

PAGE 4

CHAPTER 1  
REQUIREMENTS

PDP-11 FAMILY CENTRAL PROCESSOR WITH 4K MEMORY WITH UP TO 64 TMO3/TE16  
CONTROLLER/MAGTAPE STATIONS.

1.1 OPTIONAL EQUIPMENT USED

1. NONE

1.2 STORAGE

PROGRAM LOADS AND RUNS IN THE FIRST 4K OF MEMORY.

1.3 PRELIMINARY PROGRAMS (TO ASSURE HARDWARE OPERATION)

MAINDEC-11-DZTEA CONTROL LOGIC TEST(PART 1)  
MAINDEC-11-DZTEC BASIC FUNCTION TEST

GO1

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 5  
DZTEEA.P11 31-MAR-77 00:00

TMO3 DRIVE FUNCTION TIMER  
LOADING AND STARTING PROCEDURE

PAGE 5

CHAPTER 2  
LOADING AND STARTING PROCEDURE

2.0 LOAD & START PROCEDURE:

LOAD PROGRAM USING THE ABSOLUTE LOADER  
LOAD ADDRESS = 200  
SET OPERATING SWITCHES SEE CHAPT 3 SWITCH SETTINGS  
PRESS START

THE PROGRAM WILL THEN REQUEST THE FIRST BUS ADDRESS OF THE RHXX  
CONTROLLER, TMO3 DRIVES TO BE TESTED, TE16 SLAVES TO BE TESTED,  
AND IF SPEED TESTS ARE TO BE RUN. IN ADDITION TO EACH REQUEST A  
DEFAULT ANSWER WILL BE TYPED. TO INVOKE THE DEFAULT TYPE A  
CARRIAGE RETURN.

THE REQUESTS & THEIR DEFAULTS ARE:

TYPE FIRST ADDRESS OF CONTROLLER:172440  
TYPE TMO3 DRIVE #'S TO BE TESTED:ALL  
FOR TMO3 DRIVE X-TYPE SLAVE #'S TO BE TESTED:ALL  
SPEED TESTS ?(YES/NO):NO

NOTES: SLAVE #'S ARE NOT REQUESTED IF DEFAULT TO DRIVE REQUEST  
IS INVOKED.

IF MORE THAN 1 DRIVE OR SLAVE IS TO BE TESTED, TYPE  
BETWEEN EACH DRIVE OR SLAVE # TO BE TESTED.

SPEED TESTS CAN & WILL ONLY BE RUN BY ANSWERING YES TO THE REQUEST.

TYPE CONTROL U (↑U) TO DELETE LINE TYPED;TYPE 'RUBOUT' TO DELETE LAST  
CHARACTER(S).  
PROGRAM WILL REPORT ERRORS, AND END OF PASS.

2.1 RESTART PROCEDURE



H01

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 6  
DZTEEA.P11 31-MAR-77 00:00

THE PROGRAM MAY BE RESTARTED USING START UP PARAMETERS AT ADDRESS 210.

THE PROGRAM MAY ALSO BE RESTARTED BY TYPING A CONTROL C (↑C).  
A ↑C RESTART WILL REQUEST PARAMETERS.

NOTE: AFTER RESTARTING THE SWITCH REGISTER SHOULD  
BE SET TO PROGRAM SWITCH SETTINGS. IF 210  
IS LEFT AS THE SWITCH SETTING THE PROGRAM  
WILL SELECT & RUN TEST 10 ONLY. SEE SWITCH  
SETTINGS FOR EXPLANATION.

## 2.2 AUTOMATIC MODE OPERATION

IF THE PROGRAM IS LOADED AND RUN IN AUTOMATIC (CHAIN) MODE  
DEFAULT RESPONSES TO OPERATOR REQUESTS ARE USED, AND ALL AVAIL-  
ABLE TMO3/TE16 COMBINATIONS ARE TESTED. ADDITIONALLY THE SOFTWARE  
SWR IS INVOKED WITH A SWITCH SETTING OF 100000 IF LOADED VIA ACT11.  
NO OPERATOR INTERVENTION IS REQUIRED

\*\* EXCEPTION: IF LOADED VIA TMDP TMO3 DRIVE 0 TE16 SLAVE 0 IS  
NOT TESTED.



TM03 DRIVE FUNCTION TIMER  
SWITCH SETTINGS

PAGE 6

CHAPTER 3  
SWITCH SETTINGS

## CONTROL:

- 1) CONTROL G (↑G):  
SELECTS THE SOFTWARE SWR AND ALLOWS NEW SWITCH SETTINGS.  
THE MACHINE WILL THEN TYPE: SWR=XXXXXXXNEW=  
WHERE: XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWR.  
AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE  
OF THE FOLLOWING AT THE TTY:  
A) TYPE A NUMBER TO BE LOADED INTO THE SOFTWARE SWR.  
B) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH  
REGISTER CONTENTS WILL NOT BE CHANGED.
- 2) CONTROL A (↑A):  
ALTERNATES USAGE OF THE SWR BETWEEN HARDWARE & SOFTWARE.
- 3) CONTROL C (↑C):  
RESTARTS PROGRAM AT 200
- 4) CONTROL U (↑U):  
DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST.

|                  |                                   |   |
|------------------|-----------------------------------|---|
| SW15<br>(100000) |                                   | HALT ON ERROR THIS SWITCH WHEN SET WILL HALT THE PROCESSOR WHEN AN ERROR IS DETECTED.   |
|                  |                                   | THE PC+2 AND PSW AT THE TIME OF THE ERROR IS STORED ON THE STACK. PRESSING CONTINUE WILL CAUSE THE ERROR TO BE TYPED (IF SELECTED) AND FURTHER TESTING RESUMED. |
| SW14<br>(040000) | LOOP SUBTEST                      | THIS SWITCH WHEN SET LOOPS THE CURRENT SUBTEST REGARDLESS OF ERROR CONDITION.   |
| SW13<br>(020000) | INHIBIT ERROR TYPEOUT             | THIS SWITCH WHEN SET INHIBITS ERROR TYPEOUT.  |
| SW11<br>(004000) | INHIBIT SUB-TEST ITERATION        | THIS SWITCH WHEN SET CAUSES EACH SUBTEST TO BE EXECUTED ONLY ONCE.  |
| SW10<br>(002000) | INHIBIT FUNCTION TIME PUBLICATION | THIS SWITCH WHEN SET WILL INHIBIT THE PRINTING OF THE FUNCTION TIMES. (SEE CHAPTER 8.)  |
| SW09<br>(001000) | RING BELL ON ERROR                | THIS SWITCH WHEN SET WILL RING THE BELL ON THE TTY WHEN AN ERROR IS DETECTED.   |
| SW08<br>(000900) | PRINT TIME                        | THIS SWITCH WHEN SET WILL PRINT A TIME LINE AFTER EACH ITERATION.   |
| SW06<br>(000100) | CONTINUOUS CYCLE                  | THIS SWITCH WHEN SET WILL CAUSE THE PROGRAM TO RUN CONTINUOUSLY UNTIL STOPPED BY THE OPERATOR.  |
| SW5-0            | TEST SELECT                       | RUN SUBTEST SELECTED  |

NOTE: A TEST CAN ONLY BE SELECTED DURING STARTUP (OR RESTART).  
DO NOT INHIBIT SUBTEST ITERATIONS WHEN PROGRAM IS RUNNING.

K01

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 9  
DZTEEA.P11 31-MAR-77 00:00

TMO3 DRIVE FUNCTION TIMER  
ERRORS

PAGE 7

## CHAPTER 4

### ERRORS

TWO TYPES OF ERRORS ARE DETECTED BY THIS PROGRAM, HARDWARE ERRORS AND INCORRECT FUNCTION TIMES.

4.1 ERROR TYPEOUT FORMAT (HARDWARE): DATA RELATED ERRORS (IE: PARITY ERROR) ARE PRINTED AS SOFT ERRORS AND HAVE NO EFFECT ON TIME.

TEST # XXXXXX DEVICE ERROR

|        |        |       |       |        |        |       |
|--------|--------|-------|-------|--------|--------|-------|
| CS1    | ME     | BA    | FC    | CS2    | DS     | ER1   |
| AAAAAA | BBBBBB | CCCCC | DDDDD | EEEEEE | FFFFFF | GGGGG |

WHERE:

XXXXXX = TEST NUMBER  
AAAAAA-IIIIII = CONTENTS OF TAPE REGISTER 172440-172454

4.2 ERROR TYPEOUT FORMAT (FUNCTION TIME OUT OF RANGE)

TEST # XXXXXX OUT OF RANGE ERROR

RANGE = <AAAAAA-BBBBBB> ACTUAL = CCCCCC

TMO3 DRIVE FUNCTION TIMER  
SUBROUTINE ABSTRACTS

PAGE 8

CHAPTER 5  
SUBROUTINE ABSTRACTS

## 5.1 .SCOPE

THE SCOPE ROUTINE IS CALLED BY THE SCOPE (EMT) INSTRUCTION AT THE START OF EACH SUBTEST. THE .SCOPE ROUTINE PERFORMS THE FOLLOWING FUNCTIONS:

1. LOADS R5 WITH BASE ADDRESS
2. TYPES TIME LINE <SW08>=1
3. PROVIDES CONTINUOUS LOOP <SW14>
4. MOVES FUNCTION TIME INTO TABLE
5. OUTPUTS LINE ITEM <SW10>=1
6. DELAYS 350MS BEFORE STARTING TEST
7. INIT'S DRIVE/SLAVE
8. CLEARS THE ERROR FLAG (ERFLG)
9. CHECK FOR CONTROL G (↑G)

THE ROUTINE MONITORS SW14, SW11, SW10, SW08, AND SW07.

## 5.2 PUBLISH

THE PUBLISH ROUTINE IS CALLED FROM THE SCOPE ROUTINE IF SW10 IS EQUAL TO 0 (PUBLISH TIME DOCUMENT). THE ROUTINE WILL PRINT A THE TIME RECORDED BY THE SUBTEST.

TMO3 DRIVE FUNCTION TIMER  
SUBROUTINE ABSTRACTS

PAGE 9

## 5.3 .HLT

THE HLT ROUTINE IS CALLED BY THE HLT (TRAP) INSTRUCTION WHEN AN ERROR IS DETECTED. A HLT (TRAP) INSTRUCTION FORMATS THE ERROR INFORMATION AS SHOWN IN SEC 4.1. A HLT+1 (TRAP+1) FORMATS THE ERROR AS SHOWN IN SEC 4.2.

NO1

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 12  
DZTEEA.P11 31-MAR-77 00:00

TMO3 DRIVE FUNCTION TIMER  
MISCELLANEOUS

PAGE 10

CHAPTER 6  
MISCELLANEOUS

6.1 STACK POINTER

THE STACK POINTER IS INITIALLY SET TO 500.

6.2 EXECUTION TIME

WHEN SW11=1 (INHIBIT ITERATIONS) THE TIME REQUIRED IS 2 MIN.

WHEN SW11=0 (ITERATE SUBTESTS) THE TIME REQUIRED IS 9 MIN.

TMO3 DRIVE FUNCTION TIMER  
PROGRAM DESCRIPTION

CHAPTER 7  
PROGRAM DESCRIPTION

7.1 SAMPLE TIME DOCUMENT

TYPE FIRST ADDRESS OF CONTROLLER:172440  
TYPE TMO3 DRIVE #'S TO BE TESTED:ALL 0  
FOR TMO3 DRIVE 0- TYPE SLAVE #'S TO BE TESTED:ALL 0  
TAPE SPEED TESTS ONLY? (YES/NO):NO  
\*\*\*\*\*  
\* TMO3 DRIVE FUNCTION TIMES- DRIVE # 0 SLAVE # 7 9 CHAN. SER. # 5009  
\*

| * FUNCTION              | TIME(SPECIFICATION)   | TIME(ACTUAL)   |
|-------------------------|-----------------------|----------------|
| * WRITE FROM BOT        | RANGE=<176000-172000> | ACTUAL=174740  |
| * WRITE START           | RANGE=<009000-008700> | ACTUAL=009120  |
| * WRITE SHUTDOWN        | RANGE=<000000-007500> | ACTUAL=008840  |
| * WRITE SETTLEDOWN      | RANGE=<013000-007300> | ACTUAL=010970  |
| * READ FROM BOT         | RANGE=<045000-041000> | ACTUAL=043580  |
| * READ START            | RANGE=<000000-002400> | ACTUAL=002740  |
| * READ SHUTDOWN         | RANGE=<004100-003050> | ACTUAL=004360  |
| * READ SETTLEDOWN       | RANGE=<013000-007300> | ACTUAL=010970  |
| * READ REV START        | RANGE=<000000-002400> | ACTUAL=002740  |
| * READ REV SHUTDOWN     | RANGE=<003700-003300> | ACTUAL=003520  |
| * READ REV SETTLEDOWN   | RANGE=<013500-007300> | ACTUAL=010970  |
| * TURN AROUND DELAY F-R | RANGE=<016700-010700> | ACTUAL=013600  |
| * TURN AROUND DELAY R-F | RANGE=<016700-010700> | ACTUAL=013660  |
| * GAP SIZE-STOP HALF    | RANGE=<012900-009500> | ACTUAL=012200  |
| * GAP SIZE-START HALF   | RANGE=<011800-008500> | ACTUAL=010520  |
| * GAP SIZE-INTERRECORD  | RANGE=<014300-012600> | ACTUAL=014500  |
| * GAP CONSISANCY        | RANGE=<013500-011800> | ACTUAL=013040- |
| * DATA TIME-800BPI      | RANGE=<024000-022000> | ACTUAL=023400  |
| * DATA TIME-1600BPI     | RANGE=<025100-024100> | ACTUAL=024470  |
| * ERASE GAP TIME        | RANGE=<101000-098000> | ACTUAL=099510  |
| * WRITE FILE MARK       | RANGE=<104000-102000> | ACTUAL=103990  |



C02

DZTEE-A TM03/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 14  
DZTEEA.P11 31-MAR-77 00:00

TM03 DRIVE FUNCTION TIMER

PAGE 12

7.1.1 SAMPLE TIME DOCUMENT FOR TAPE SPEED TESTS

TYPE FIRST ADDRESS OF CONTROLLER 172440:

TYPE TM03 DRIVE #'S TO BE TESTED:ALL 0

FOR TM03 DRIVE 0- TYPE SLAVE #'S TO BE TESTED:ALL 7

SPEED TESTS ONLY? (YES/NO):NO Y

\*\*\*\*\*  
\*TM03 DRIVE FUNCTION TIMES- DRIVE # 0 SLAVE # 7 9 CHAN. SER. # 5009

| *FUNCTION       | TIME(SPECIFICATION)   | TIME(ACTUAL)  |
|-----------------|-----------------------|---------------|
| *TAPE SPEED FWD | RANGE=<022700-021700> | ACTUAL=022500 |
| *TAPE SPEED REV | RANGE=<022700-021700> | ACTUAL=022500 |

TMO3 DRIVE FUNCTION TIMER

7.2 TEST SEQUENCE WITH RELATED ADJUSTMENTS AND ASSOCIATED HARDWARE

| TEST NO./NAME               | RELATED ADJUSTMENTS              | ASSOCIATED HARDWARE        |
|-----------------------------|----------------------------------|----------------------------|
| 1. WRITE FROM BOT           | *NONE                            | *M8911 ROM*M8903 ACCL CNTR |
| 2. WRITE START              | * "                              | * " * "                    |
| 3. WRITE SHUTDOWN           | * "                              | * " * "                    |
| 4. WRITE SETTLEDOWN         | * "                              | *M8910 SETTLEDOWN ONE SHOT |
| 5. READ FROM BOT            | * "                              | *M8911 ROM*M8903 ACCL CNTR |
| 6. READ START               | * "                              | * " * "                    |
| 7. READ SHUTDOWN            | * "                              | * " * "                    |
| 10. READ SETTLEDOWN         | * "                              | *M8910 SETTLEDOWN ONE SHOT |
| 11. READ REVERSE START      | * "                              | *M8911 ROM*M8903 ACCL CNTR |
| 12. READ REVERSE SHUTDOWN   | * "                              | * " * "                    |
| 13. READ REVERSE SETTLEDOWN | * "                              | *M8910 SETTLEDOWN ONE SHOT |
| 14. TURN AROUND F-R         | * "                              | *M8911 ROM*M8903 ACCL CNTR |
| 15. TURN AROUND R-F         | * "                              | * " * "                    |
| 16. GAP SIZE-STOP HALF      | *FWRD/REV SPEED-START/STOP-RAMPS | *CAPSTAN SERVO LOOP        |
| 17. GAP SIZE-START HALF     | *SAME AS IN TEST 16              | * " " "                    |
| 20. GAP SIZE INTERRECORD    | *FWD/REV SPEED                   | * " " "                    |

TMO3 DRIVE FUNCTION TIMER

|                        |                     |                            |
|------------------------|---------------------|----------------------------|
| 21. GAP CONSISTENCY    | *SAME AS IN TEST 16 | *WRITE CLOCK               |
| 22. DATA TIME 800 BPI  | *NONE               | * " "                      |
| 23. DATA TIME 1600 BPI | * "                 | * " "                      |
| 24. ERASE GAP TIME     | * "                 | *M8911 ROM*M8903 ACCL CNTR |
| 25. WRITE FILE MARK    | * "                 | * " " * " " "              |
| 26. TAPE SPEED-FORWARD | *FWD SPEED          | *CAPSTAN SERVO LOOP        |
| 27. TAPE SPEED-REVERSE | *REVERSE SPEED      | *CAPSTAN SERVO LOOP        |

\*\*\*\*\*NOTE: IF TIME PROBLEMS APPEAR IN T1 THRU T25, RUN TAPE SPEED TESTS FIRST\*\*\*\*\*  
TEST 26 & 27 REQUIRE AN 800 BPI SKEW TAPE

## TM03 DRIVE FUNCTION TIMER

PAGE 15

## 7.3 SUBTEST DESCRIPTIONS:

THE FIRST THIRTEEN (13) TESTS (T1 - T15) ARE CHECKS OF THE ROM CIRCUITS IN THE TE16 (M9811), THE ACCL COUNTER IN THE TM03 (M9903), AND THE SETTLEDOWN ONE SHOT (M9910).

## T1. WRITE FROM BOT:

THIS TEST WILL MEASURE ACCELERATION DELAY REQUIRED TO MOVE THE TAPE APPROXIMATELY SEVEN (7) INCHES FORWARD FROM DEAD STOP AT BOT BEFORE STARTING TO TRANSFER DATA.

1. ASSURE TAPE IS STOPPED AT BOT.
2. ISSUE A WRITE COMMAND
3. MONITOR BIT 15 OF TC (ACCL)
4. TIME FROM GO TO ACCL RESET IS BOT DELAY
5. STOP

## T2. WRITE START:

THIS TEST WILL MEASURE ACCELERATION DELAY JUST AS IN T1. HOWEVER THE TIME WILL BE LESS WHEN NOT STARTING FROM BOT.

1. LEAVE TAPE AT ITS PRESENT POSITION. ASSURE THAT IT IS STOPPED
2. ISSUE A WRITE COMMAND
3. MONITOR BIT 15 OF TC (ACCL)
4. TIME FROM GO TO RESET OF ACCL IS START DELAY
5. STOP

## T3. WRITE SHUTDOWN:

THIS TEST WILL MEASURE THE TIME FROM EOR (LAST CHARACTER WRITTEN ON TAPE) TO THE START OF SETTLEDOWN TIME. THIS ASSURES, IN PART, A PROPER INTERRECORD GAP.

1. LEAVE TAPE AT ITS PRESENT POSITION. ASSURE THAT IT IS STOPPED
2. ISSUE A WRITE COMMAND.
3. MONITOR FRAME COUNTER AND BIT 4 OF DS (SDWN)
4. TIME FROM FC=0 TO ASSERTION OF SDWN IS THE SHUTDOWN TIME.
5. STOP

## T4. WRITE SETTLEDOWN:

THIS TEST WILL MEASURE THE SLOWDOWN TIME. THE TIME FROM THE START OF SLOWDOWN UNTIL THE TAPE SHOULD BE STOPPED. THIS IS A PART OF THE GAP TIMING IN LOGIC. THE MECHANICAL POSITIONING OF THE TAPE IN THE GAP DISTANCE WILL BE MEASURED IN A LATER TEST.

1. LEAVE TAPE AT ITS PRESENT POSITION. ASSURE THAT IT IS STOPPED
2. ISSUE A WRITE COMMAND
3. MONITOR BIT 4 OF DS (SDWN)
4. TIME FROM SET OF SDWN TO RESET OF SDWN IS THE SETTLEDOWN DELAY
5. STOP

## T5. READ FROM BOT

THIS MEASUREMENT IS MADE EXACTLY AS THE WRITE MEASUREMENT IN T1. USE THE SAME RECORD THAT WAS WRITTEN IN T1.

1. REWIND TO BOT
2. ASSURE TAPE HAS HAD TIME TO COME TO A COMPLETE STOP
3. READ FORWARD 1 RECORD.
4. MONITOR BIT 15 OF TC (ACCL)
5. TIME FROM GO TO ACCL IS BOT DELAY
6. STOP

## T6. READ START

THIS TEST MEASURES THE SAME DELAY AS IN T2.

1. WRITE 1 RECORD, THEN BACKSPACE OVER IT, ASSURE TAPE IS STOPPED.
2. ISSUE A READ FORWARD OF THE RECORD WRITTEN IN STEP 1.
3. MONITOR BIT 15 OF TC (ACCL)
4. TIME FROM GO TO RESET OF ACCL IS START DELAY
5. STOP

## T7. READ SHUTDOWN:

THIS TEST MEASURES THE SAME DELAY AS IN T3.

1. WRITE 1 RECORD, THEN BACKSPACE OVER IT, ASSURE TAPE IS STOPPED.
2. READ FORWARD THE RECORD WRITTEN IN STEP 1.
3. MONITOR FRAME COUNT AND BIT 4 OF DS (SDWN).
4. TIME FROM FC=RECORD SIZE (LAST FRAME READ) TO SDWN=1 IS THE SHUTDOWN TIME.
5. STOP

## T10. READ SETTLEDOWN:

THIS TEST MEASURES THE SAME DELAY AS IN T4.

1. WRITE 1 RECORD, THEN BACKSPACE OVER IT, ASSURE TAPE IS STOPPED.
2. READ FORWARD THE RECORD WRITTEN IN STEP 1.
3. MONITOR BIT 4 OF DS (SDWN)
4. TIME FROM SET OF SDWN TO RESET OF SDWN IS THE SETTLEDOWN DELAY.
5. STOP

## TMO3 DRIVE FUNCTION TIMER

PAGE 17

## T11. READ REVERSE START:

THIS TEST WILL MEASURE THE START DELAY IN THE REVERSE DIRECTION.

1. WRITE 1 RECORD, ASSURE TAPE IS STOPPED.
2. READ REVERSE THE RECORD WRITTEN IN STEP 1.
3. MONITOR BIT 15 OF TC (ACCL)
4. THE TIME FROM GO TO RESET OF ACCL IS THE START TIME
5. STOP

## T12. READ REVERSE SHUTDOWN

THIS TEST WILL MEASURE THE READ SHUTDOWN IN THE REVERSE DIRECTION.

1. WRITE 1 RECORD, ASSURE TAPE IS STOPPED.
2. READ REVERSE THE RECORD WRITTEN IN STEP 1.
3. MONITOR FRAME COUNTER AND BIT 4 OF DS (SDWN).
4. TIME FROM FC=RECORD SIZE (LAST FRAME READ) TO SDWN=1 IS THE READ REVERSE SHUTDOWN TIME.
5. STOP

## T13. READ REVERSE SETTLEDOWN:

THIS TEST WILL MEASURE THE READ SETTLEDOWN IN THE REVERSE DIRECTION.

1. WRITE 1 RECORD, ASSURE TAPE IS STOPPED.
2. READ REVERSE THE RECORD WRITTEN IN STEP 1.
3. MONITOR BIT 4 OF DS (SDWN)
4. TIME FROM SET OF SDWN TO RESET OF SDWN IS THE SETTLEDOWN DELAY
5. STOP

## T14. TURN AROUND DELAY-FORWARD TO REVERSE

THIS TEST WILL MEASURE THE TIME REQUIRED FOR THE TAPE TO CHANGE DIRECTION.

1. LEAVE TAPE AT ITS PRESENT POSITION. ASSURE THAT IT IS STOPPED
2. ISSUE A WRITE FORWARD OF AT LEAST 20 FRAMES
3. MONITOR BIT 7 OF DS (DRY)
4. WHEN DRY IS ASSERTED (EOR), IMMEDIATELY ISSUE A READ REVERSE OF THAT RECORD.
5. MONITOR BIT 15 OF TC (ACCL).
6. TIME FROM GO OF READ REVERSE TO RESET OF ACCL IS THE TURNAROUND TIME.
7. STOP



## TMO3 DRIVE FUNCTION TIMER

PAGE 18

T15. TURN AROUND DELAY-REVERSE TO FORWARD

THIS TEST WILL MEASURE THE TIME AS IN T14, BUT IN THE  
OPPOSITE DIRECTION.

1. WRITE 1 RECORD.
2. ASSURE TAPE IS STOPPED
3. READ REVERSE
4. MONITOR DRY (BIT 7 OF DS)
5. WHEN DRY = 1, ISSUE A READ FORWARD
6. MONITOR ACCL (BIT 15 OF TC)
7. TIME FROM GO FORWARD TO ACCL = 1 IS THE TURN AROUND TIME.
8. STOP.

## TMO3 DRIVE FUNCTION TIMER

PAGE 19

## GAP MEASUREMENTS:

THE PREVIOUS THIRTEEN (13) TESTS WERE MEASUREMENTS OF LOGIC DELAYS PERFORMED BY THE TMO3 OR TE16 IN ORDER TO ALLOW FOR PROPER ACCELERATION AND DECELERATION OF TAPE ACCORDING TO THE DESIRED INTERCORD GAP (.5 INCHES). THIS TEST, HOWEVER, WILL MEASURE THE PHYSICAL SIZE OF THE INTERCORD GAP THAT EXISTS ON TAPE AS A RESULT OF THE START/STOP TIMES OF THE CAPSTAN ITSELF. BECAUSE THE INTERCORD GAP IS CREATED BY TWO ACTIONS, THE START OF MOTION AND THE STOP OF MOTION IT IS NECESSARY TO MAKE TWO SEPERATE MEASUREMENTS. A THIRD MEASUREMENT, MADE ON THE FLY, OF THE ENTIRE LENGTH OF THE GAP WILL ALSO BE MADE.

## T16. GAP SIZE (STOP HALF)

THIS TEST WILL MEASURE THE DISTANCE TRAVLED BY THE TAPE IN A STOP CYCLE. IN OTHER WORDS, THE DISTANCE INTO THE IRG.

1. WRITE 1 RECORD.
2. ASSURE TAPE IS STOPPED.
3. ISSUE A READ REVERSE OVER THE RECORD
4. MONITOR THE FRAME COUNT FOR THE FIRST FRAME READ (FC = 1)
5. THE TIME FROM GO=1 TO FC=1 IS THE LENGTH OF THE GAP
6. STOP

## T17. GAP SIZE (START HALF)

THIS TEST WILL MEASURE THE DISTANCE OF TAPE TRAVEL DURING START UP.

1. WRITE 1 RECORD, THEN REVERSE OVER IT, ASSURE TAPE IS STOPPED.
2. ISSUE A READ FORWARD
3. MONITOR FC FOR FC=1
4. TIME FROM GO=1 TO FC=1 IS START DISTANCE
5. STOP

## T20. GAP SIZE (INTERRECORD)

THIS TEST WILL MEASURE THE ENTIRE LENGTH OF THE IRG ON THE FLG. THE TIME VALUE OF THIS TEST SHOULD NOT BE EQUAL TO A SUMMATION OF T16 AND T17 DUE TO THE FACT THAT THE ACCELERATION AND DECELERATION CURVES ARE NOT IN EFFECT. THE VALUE HERE SHOULD ACTUALLY BE LESS THAN THE SUM OF T16 AND T17.

1. WRITE 2 RECORDS.
2. READ REVERSE OVER THE SECOND RECORD
3. MONITOR DRY (BIT 7 OF DS)
4. WHEN DRY = 1, ISSUE A SECOND READ REVERSE
5. MONITOR FRAME COUNT
6. TIME FROM GO=1 OF SECOND READ REVERSE TO FC=1 IS THE LENGTH OF THE GAP.

L02

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 23  
DZTEEA.P11 31-MAR-77 00:00

7. STOP

## TMO3 DRIVE FUNCTION TIMER

PAGE 20

## T21. GAP CONSISTENCY:

NOW THAT WE HAVE ESTABLISHED THAT THE INTERRECORD GAP IS THE PROPER SIZE, LET US DETERMINE THE CONSISTENCY OF THE GAP UNDER VARIOUS COMMAND EXECUTION TIMES. BY WRITING A SERIES OF RECORDS, EACH WITH A DIFFERENT DELAY BETWEEN EXECUTION, WE CAN ESTABLISH THE CONSISTENCY OF THE GAPS BY READING THESE RECORDS AND MONITORING THEIR INTERRECORD GAPS, ON THE FLY.

1. REWIND TAPE TO BOT.
2. WRITE ONE (1) RECORD TO GET TAPE OFF BOT
3. WRITE SIXTEEN (16) RECORDS WITH A PROGRESSIVE DELAY OF FROM 0 TO 16 MILLISECOND (APPROX) BETWEEN COMMANDS.
4. BACKSPACE 16 RECORDS AND ALLOW THE TAPE TO STOP.
5. READ FORWARD (NON-STOP) OVER THESE 16 RECORDS, EACH TIME MONITORING THE TIME FROM THE END OF RECORD (DRY) UNTIL THE FRAME COUNT NEXT GOES FROM 0 TO 1 (FC=1).
6. THE TIMES FROM DRY TO FC=1 IS THE GAP TIME AND IT SHOULD REMAIN CONSISTANT FOR ALL RECORDS.
7. STOP

\*\*(SEE GTIMTBL IN LISTING FOR GAP TIMES)\*\*

## T22. DATA TIME AT 800 BPI:

THIS TEST WILL MEASURE THE TIME REQUIRED TO WRITE ONE (1) INCH OF TAPE AT 800 BPI. BY WRITING A RECORD OF ENOUGH FRAMES TO MOVE THE TAPE 1 INCH (800 FRAMES), DATA RATE CAN BE VARIFIED.

1. REWIND TO BOT AND ALLOW TAPE TO STOP
2. WRITE A RECORD AT 200 BPI.
3. MONITOR DRY (BIT 7 OF DS) FOR EACH RECORD
4. THE TIME FROM FC=FC+1 TO DRY WILL BE THE TIME REQUIRED FOR 1 INCH AT THE SELECTED DENSITY
5. STOP

T23. DATA TIME AT 1600 BPI (PE):  
REPEAT STEPS 1 THRU 5 AT 1600 BPI.

## TM03 DRIVE FUNCTION TIMER

PAGE 21

## T24. ERASE:

THE ERASE COMMAND WILL CAUSE AN AREA OF THE THREE (3) INCHES TO BE DC ERASED IN THE FORWARD DIRECTION. THIS TEST WILL ASSURE THAT THE PROPER DISTANCE IS ERASED.

1. LEAVE TAPE AT ITS PRESENT POSITION.
2. ISSUE AN ERASE COMMAND.
3. MONITOR DRY (BIT 7 OF DS)
4. THE TIME FROM GO TO DRY WILL BE THE TIME REQUIRED TO ERASE 3 INCHES OF TAPE AND WILL REFLECT THE DISTANCE. DENSITY IS NOT A FACTOR.
5. STOP

## T25. TAPE MARK:

THIS TEST IS ALSO A CHECK ON THE THREE (3) INCH GAP. WHEN A TAPE MARK IS WRITTEN, A 3 INCH GAP IS CREATED BEFORE DATA IS PUT ON TAPE.

1. LEAVE TAPE AT ITS PRESENT POSITION
2. ISSUE A WRITE TAPE MARK COMMAND
3. MONITOR DRY (BIT 7 OF DS)
4. THE TIME FROM GO TO DRY WILL BE THE TIME REQUIRED TO WRITE THE TM RECORD PLUS THE 3 INCH GAP.
5. STOP

TMO3 DRIVE FUNCTION TIMER

T26. TAPE SPEED FORWARD:

THIS TEST REQUIRES THE USE OF AN 800 BPI SKEW TAPE!  
THE OPERATOR WILL BE REQUIRED TO MOUNT THE SKEW TAPE  
BEFORE EXECUTING THE TEST. THE SKEW TAPE IS THE ONLY  
WAY TO ASSURE THAT TAPE IS MOVING AT THE PROPER SPEED  
BECAUSE THE FREQUENCY OF FRAMES ON A SKEW TAPE IS  
GUARANTEED TO BE ACCURATE.

1. ASSURE TAPE IS STOPPED AT BOT.
2. ISSUE A READ FORWARD (800 BPI, NORMAL)
3. MONITOR FC FOR FC = 800(10)
4. MONITOR FC FOR FC = 8800(10)
5. TIME FROM FC = 800 TO FC = 8800 IS THE TIME REQUIRED FOR TAPE TO TRAVEL 10 INCHES
6. DIVIDE THE TIME FOR 10 INCHES BY 10.
7. THE RESULT IS AN AVERAGE SPEED FOR 1 INCH.
8. STOP.

T27. TAPE SPEED REVERSE:

THIS TEST IS THE SAME AS TEST 31, BUT SPEED IS MEASURED IN THE REVERSE DIRECTION.

1. ADVANCE TAPE OFF OF BOT.
2. ISSUE A READ REVERSE.
3. REPEAT STEPS 3 THRU 6 IN THE REVERSE DIRECTION.
4. STOP.

%

981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004

100000  
040000  
020000

```

.LIST BIN,LOC,SEQ

.NLIST MC
.NLIST TOC
.LIST ME
.ENABLE ABS,AMA
.MCALL SCPVEC,SCPREG,SCATCH,STYPE,SACT11,SEOP,SCHAIN
.TITLE DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER
.SBTTL STARTING INSTRUCTIONS
;LOADING AND STARTING PROCEDURE
LOAD PROGRAM USING ABS LOADER
LOAD ADDRESS 200
SET SWITCH OPTIONS
PRESS START

;RESTART PROCEDURE
LOAD ADDRESS 210
SET SWITCH OPTIONS
PRESS START

;SWITCH REGISTER SWITCH ASSIGNMENTS
SW15= 100000 ;HALT ON ERROR
SW14= 040000 ;LOOP SUBTEST
SW13= 020000 ;INHIBIT ERROR TYPE OUT

```

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER  
DZTEEA.P11 31-MAR-77 00:00

MACY11 30(1046) 27-JUN-77 17:03 PAGE 27  
STARTING INSTRUCTIONS

```

1005      004000      SW11= 004000      ;INHIBIT SUBTEST ITERATION
1006      002000      SW10= 002000      ;INHIBIT PUBLISHING TIME SPECIFICATION
1007      001000      SW09= 001000      ;RING BELL ON ERROR
1008      000400      SW08= 000400      ;TYPE LINE ITEM AFTER EACH ITERATION
1009      000200      SW07= 00200      ;NOT USED
1010      000100      SW06= 000100      ;CONTINUOUS CYCLE
1011      :           SW05-SW00      ;RUN TEST SELECTED
1012      :           **NOTE: IF <SW15-SW00> = 177777 AT STARTUP USE SOFTWARE
1013      :           SWITCH REGISTER.
1014
1015      ;CONSOLE COMMANDS
1016      :           CONTROL C      ;RESTART PROGRAM (SAME AS START @ 200)
1017      :           CONTROL G      ;SET NEW SOFTWARE SWITCH REGISTER
1018      :           CONTROL U      ;DELETE LINE TYPED
1019      :           RUBOUT (DELETE) ;DELETE LAST CHAR TYPED
1020
1021      ;GENERAL REGISTER USAGE:
1022      :           R0=ADDRESS OF 'FC' REGISTER (SET BY SCOPE)
1023      :           R1=ADDRESS OF 'DS' REGISTER (SET BY SCOPE)
1024      :           R2=RETURN PC FROM TIMER (SET BY EACH TEST)
1025      :           R3=INDEX INDICATING PREVIOUS OSCILLATOR POLARITY (SET BY TIMER)
1026      :           R4=CONTAINS 'TICK' COUNT WHEN TIMER IS RUNNING (SET BY TIMER)
1027      :           R5=ADDRESS OF CSI (SET BY SCOPE)
1028
1029      .SBTTL MACRO DEFINITIONS
1030      .MACRO SAVE
1031      JSR PC,SAVE      ;SAVE REGISTERS ON THE STACK
1032      .ENDM SAVE
1033      .MACRO RESTORE
1034      JSR PC,RESTORE  ;RESTORE REGISTERS FROM THE STACK
1035      .ENDM RESTORE
1036      .MACRO INPUT
1037      JSR PC,INPUT    ;GET USER INPUT
1038      .ENDM INPUT
1039      .MACRO REWIND
1040      JSR PC,REWIND   ;REWIND SLAVE
1041      BVS 99$         ;BRANCH IF ERROR ON REWIND
1042      .ENDM REWIND
1043      .MACRO TIMEON
1044      JSR PC,TIMON    ;TURN TIMER ON
1045      .ENDM TIMEON
1046      .MACRO TIMCHK
1047      JMP TIMER(R3)  ;GO TO TIMER & RETURN VIA R2
1048      .ENDM TIMCHK
1049      .MACRO SETGO
1050      INC (R5)       ;SET 'GO' BIT
1051      .ENDM SETGO
1052
1053      .SBTTL REGISTER ASSIGNMENTS
1054      ;;DEFINITIONS AND REGISTER ASSIGNMENTS
1055      ;;GENERAL REGISTER ASSIGNMENTS
1056      000000      R0=%0
1057      000001      R1=%1
1058      000002      R2=%2
1059      000003      R3=%3
1060      000004      R4=%4

```



1061 000005  
1062 000006  
1063 000007  
1064 000000  
1065 000001  
1066 000002  
1067 000003  
1068 000004  
1069 000005  
1070  
1071  
1072 177776  
1073 177774  
1074 177772  
1075 177770  
1076 177560  
1077 177562  
1078 177564  
1079 177566  
1080  
1081  
1082 000004  
1083 000010  
1084 000014  
1085 000014  
1086 000014  
1087 000020  
1088 000024  
1089 000030  
1090 000034  
1091 000060  
1092 000064  
1093 000114  
1094 000240  
1095 000244  
1096 000250  
1097  
1098  
1099 172540  
1100 000104  
1101 177546  
1102 000100  
1103 177514  
1104 177516  
1105  
1106  
1107 172440  
1108  
1109  
1110 000000  
1111 000002  
1112 000004  
1113 000006  
1114 000010  
1115 000012  
1116 000014

RS=%5  
SP=%6  
PC=%7  
R10=%0  
R11=%1  
R12=%2  
R13=%3  
R14=%4  
R15=%5

:: REGISTER ADDRESSES

PSW= 177776  
SLR= 177774  
PIRQ= 177772  
UBREAK= 177770  
TKS= 177560  
TKB= 177562  
TPS= 177564  
TPB= 177566

:: PROCESSOR STATUS WORD  
:: STACK LIMIT REGISTER (11/40,11/45)  
:: PROGRAM INTERRUPT REQ. (11/45)  
:: MICRO-BREAK REGISTER (11/45)  
:: KEYBOARD CSR  
:: KEYBOARD DATA BUFFER REGISTER  
:: TELEPRINTER CSR  
:: TELEPRINTER DATA BUFFER REGISTER

:: VECTOR ADDRESSES

ERRVEC=4  
RESVEC=10  
TBITVEC=14  
TRTVEC=14  
BPTVEC=14  
IOTVEC=20  
PFVEC=24  
EMTVEC=30  
TRAPVEC=34  
TKVEC= 60  
TPVEC=64  
PARVEC= 114  
PIRVEC=240  
FPEVEC=244  
MMVEC=250

:: ADDRESS OF ERROR VECTOR  
:: ADDRESS OF RESERVED INST. TRAP VECTOR  
:: ADDRESS OF 'I' BIT TRAP VECTOR  
:: ADDRESS OF 'TRACE' TRAP VECTOR  
:: ADDRESS OF 'BREAKPOINT' TRAP VECTOR  
:: ADDRESS OF IOT TRAP VECTOR  
:: ADDRESS OF POWER FAIL TRAP VECTOR  
:: ADDRESS OF EMT VECTOR  
:: ADDRESS OF TRAP VECTOR  
:: ADDRESS OF TTY KEYBOARD INT. VECTOR  
:: ADDRESS OF TTY PRINTER INTERRUPT VECTOR  
:: ADDRESS OF MA/MF PARITY ERROR VECTOR  
:: ADDRESS OF PIRQ VECTOR  
:: ADDRESS OF FLOATING POINT INT. VECTOR  
:: ADDRESS OF MEM MGMT ERROR TRAP VECTOR

:: CLOCK ADDRESS AND VECTORS

PLKCSR= 172540  
PLKVEC= 104  
LKS= 177546  
LKVEC= 100  
LPS= 177514  
LPB= 177516

;KW11-P  
;KW11-L  
;LP11

:: RH, TMO3/TE16 REGISTERS

TMCS1= 172440

:: TMO3/TE16 INDEX VALUES

CS1= 00  
WC= 02  
BA= 04  
FC= 06  
CS2= 10  
DS= 12  
ER= 14

;CONTROL STATUS #1  
;BUS ADDRESS REGISTER  
;FRAME COUNT  
;CONTROL STATUS #2  
;DRIVE STATUS  
;ERROR REG #1

: ATTENTION SUMMARY  
: DATA BUFFER REG  
: MAINTENANCE REG  
: DRIVE TYPE REG  
: SERIAL NUMBER REGISTER  
: TAPE CONTROL REG

1117 000016  
1118 000022  
1119 000024  
1120 000026  
1121 000030  
1122 000032

RS= 16  
DB= 22  
MR= 24  
DT= 26  
SN= 30  
TC= 32

.SBTTL TMO3/TE16 REGISTER BITS  
;RHCS1-CS1(R5)

1125 000001  
1126 000000  
1127 000002  
1128 000006  
1129 000010  
1130 000026  
1131 000024  
1132 000030  
1133 000032  
1134 000050  
1135 000056  
1136 000060  
1137 000070  
1138 000076  
1139 000100  
1140 000200  
1141 000400  
1142 001000  
1143 002000  
1144 004000  
1145 020000  
1146 040000  
1147 100000

GO= 1  
NOP= 0  
RWD OFF= 2  
RWD= 6  
DRY CLR= 10  
WFK= 26  
ERASE= 24  
SPCFWD= 30  
SPCREV= 32  
WCHKF= 50  
WCHKR= 56  
WFWO= 60  
RDFWD= 70  
RDREV= 76  
IE= 100  
RDY= 200  
A16= 400  
A17= 1000  
PSFL= 2000  
DV1= 4000  
MCPE= 20000  
TRE= 40000  
SC= 100000

;RHCS2-CS2(R5)

1149 000000  
1150 000001  
1151 000002  
1152 000003  
1153 000004  
1154 000005  
1155 000006  
1156 000007  
1157 000010  
1158 000020  
1159 000040  
1160 000100  
1161 000200  
1162 000400  
1163 001000  
1164 002000  
1165 004000  
1166 010000  
1167 020000  
1168 040000  
1169 100000

DV0= 0  
DV1= 1  
DV2= 2  
DV3= 3  
DV4= 4  
DV5= 5  
DV6= 6  
DV7= 7  
BAI= 10  
PAT= 20  
CLR= 40  
IR= 100  
OR= 200  
MOPE= 400  
MXF= 1000  
PGE= 2000  
NEM= 4000  
NED= 10000  
LPE= 20000  
WCE= 40000  
DLT= 100000

;RHDS-DS(R5)

1171 000001  
1172

SLA= 1

|      |        |                        |        |
|------|--------|------------------------|--------|
| 1173 | 000002 | BOT=                   | 2      |
| 1174 | 000004 | TMK=                   | 4      |
| 1175 | 000010 | IDB=                   | 10     |
| 1176 | 000020 | SDWN=                  | 20     |
| 1177 | 000040 | PES=                   | 40     |
| 1178 | 000100 | SSC=                   | 100    |
| 1179 | 000200 | DRY=                   | 200    |
| 1180 | 000400 | DPR=                   | 400    |
| 1181 | 002000 | EOT=                   | 2000   |
| 1182 | 004000 | WRL=                   | 4000   |
| 1183 | 010000 | MOL=                   | 10000  |
| 1184 | 020000 | PIP=                   | 20000  |
| 1185 | 040000 | ERR=                   | 40000  |
| 1186 | 100000 | ATA=                   | 100000 |
| 1187 |        | ;RHER-ER(R5)           |        |
| 1188 | 000001 | ILF=                   | 1      |
| 1189 | 000002 | ILR=                   | 2      |
| 1190 | 000004 | RMR=                   | 4      |
| 1191 |        |                        |        |
| 1192 | 000020 | FMT=                   | 20     |
| 1193 | 000100 | INCVAE=                | 100    |
| 1194 | 000200 | PEFLRC=                | 200    |
| 1195 | 000400 | MSG=                   | 400    |
| 1196 | 001000 | FCE=                   | 1000   |
| 1197 | 002000 | CSITM=                 | 2000   |
| 1198 | 004000 | NEF=                   | 4000   |
| 1199 | 010000 | DTE=                   | 10000  |
| 1200 | 020000 | OPI=                   | 20000  |
| 1201 | 040000 | UNS=                   | 40000  |
| 1202 |        |                        |        |
| 1203 |        | ;RHMR-MR(R5)           |        |
| 1204 | 000100 | OSC=                   | 100    |
| 1205 |        |                        |        |
| 1206 |        | ;RHDT-DT(R5)           |        |
| 1207 | 002000 | SPR=                   | 2000   |
| 1208 | 010000 | CH7=                   | 10000  |
| 1209 | 040000 | TAP=                   | 40000  |
| 1210 |        |                        |        |
| 1211 |        | ;RHTC-TC(R5)           |        |
| 1212 | 001700 | NORM11=                | 1700   |
| 1213 | 000320 | CDM11=                 | 320    |
| 1214 | 000000 | BPI200=                | 0      |
| 1215 | 000400 | BPI556=                | 000400 |
| 1216 | 001000 | BPI800=                | 001000 |
| 1217 | 002300 | PE1600=                | 002300 |
| 1218 | 100000 | ACCL=                  | 100000 |
| 1219 |        |                        |        |
| 1220 |        |                        |        |
| 1221 |        |                        |        |
| 1222 |        | ;INSTRUCTION EQUATES   |        |
| 1223 | 104400 | HLT=                   | TRAP   |
| 1224 | 104000 | SCOPE=                 | EMT    |
| 1225 | 000004 | TYPE=                  | IOT    |
| 1226 |        |                        |        |
| 1227 |        | ;MISCELLANEOUS EQUATES |        |
| 1228 | 005620 | OUTBUF=                | INIT   |

;OUTPUT BUFFER STARTS AT BEG OF PROGRAM

DZTEE-A TM03/TE16 DRIVE FUNCTION TIMER  
 DZTEEA.P11 31-MAR-77 00:00

MACY11 30(1046) 27-JUN-77 17:03 PAGE 31  
 TM03/TE16 REGISTER BITS

1229 177400  
 1230 177600  
 1231  
 1232 000001  
 1233 000003  
 1234 000007  
 1235 000011  
 1236 000012  
 1237 000015  
 1238 000017  
 1239 000025

FRMCNT= -256.  
 WRDCNT= -128.  
 ;ASCII EQUATES  
 CNTRLA= 1  
 CNTRLC= 3  
 CNTRLG= 7  
 HT= 11  
 LF= 12  
 CR= 15  
 CNTRLO= 17  
 CNTRLU= 25

;FRAME COUNT  
 ;WORD COUNT  
 ;ASCII CODE FOR CONTROL A (↑A)  
 ;ASCII CODE FOR CONTROL C (↑C)  
 ;ASCII CODE FOR CONTROL G (↑G)  
 ;ASCII CODE FOR HORIZONTAL TAB  
 ;ASCII CODE FOR LINE FEED  
 ;ASCII CODE FOR CARRIAGE RETURN  
 ;ASCII CODE FOR CONTROL O (↑O)  
 ;ASCII CODE FOR CONTROL U (↑U)

```

1240 ; SETUP TRAP VECTORS
1241      .=TBITVEC
1242      .WORD      +2          ; SET 'T' TRAP TO TIMER ROUTINE
1243      .WORD      HALT        ; PRIORITY LEVEL 7
1244      .WORD      .TYPE       ; SET IOT TRAP TO .TYPE ROUTINE
1245      .WORD      0           ; PRIORITY LEVEL 0
1246      .WORD      PFVEC+2     ; POWER FAIL TRAP TO HALT
1247      .WORD      HALT        ; AT PFVEC+2
1248      .WORD      .SCOPE      ; SET EMT TRAP TO .SCOPE ROUTINE
1249      .WORD      340         ; PRIORITY LEVEL 7
1250      .WORD      .HLT        ; SET TRAP TRAP TO .HLT ROUTINE
1251      .WORD      340         ; PRIORITY LEVEL 7
1252
1253 ; ACT11 HOOK *****
1254      $SVPC=.                ; SAVE CURRENT LOCATION CTR
1255      .=46
1256      .WORD      $ENDAD      ; SET LOCATION 46
1257      .=52
1258      .WORD      0           ; SET LOCATION 52 = 0
1259      .=$SVPC                ; RESTORE LOCATION CTR
1260
1261      .=TKVEC
1262      .WORD      TKISR
1263      .WORD      200
1264
1265 ; SOFTWARE SWITCH REGISTER LOC. 176
1266      .=176
1267      .WORD      SW06        ; SOFTWARE SWITCH REGISTER
1268
1269      .=200
1270      JMP      @#INIT        ; GO TO START OF PROGRAM
1271      .=210
1272      JMP      @#RSTRT      ; RESTART ADDRESS
1273
1274      .=500
1275      STKPTR= 600            ; STACK
1276
1277      .=1000
1278      .PROGRAM TAGS
1279      SWR: 177570            ; SWITCH REGISTER
1280      SCPADR: .WORD 0
1281      DRVNUM: .BYTE 0       ; TMO3 DRIVE UNDER TEST
1282      SLVNUM: .BYTE 0       ; TE16 SLAVE UNDER TEST
1283      SLVPTR: .WORD 0       ; POINTER TO SLAVE TABLE (SLVTBL) BELOW
1284      TMBASE: .WORD TMC51   ; BASE ADDRESS OF TMO3/TE16 REGISTERS
1285      ATIME: .WORD 0        ; CONTAINS 'TICK' COUNT
1286      ATIMTBL: .BLKW 16.   ; EACH ENTRY CONTAINS TIME FOR FUNCTION
1287                                     ; ENTRIES ARE MADE BY 'SCOPE' ROUTINE
1288      GAP: .BLKW 16.        ; TIMES RECORDED BY 'GAP CONSISTANCY' TEST
1289      DELTIM: .WORD 0
1290      OCTALO: .WORD 0
1291      GAP: .BYTE 0          ; CONTAINS GAP # (USED FOR TST 021)
1292      ITCNT: .BYTE 0        ; ITERATION COUNT
1293      TSTNUM: .BYTE 0       ; TEST #
1294      ERFLG: .BYTE 0        ; ERROR FLAG
1295      PRGFLG: .BYTE 0        ; PROGRAM FLAG

```

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER  
DZTEEA.P11 31-MAR-77 00:00

MACY11 30(1046) 27-JUN-77 17:03 PAGE 33  
TMO3/TE16 REGISTER BITS

|      |        |        |     |
|------|--------|--------|-----|
| 1296 | 001125 | 000    |     |
| 1297 | 001126 | 000    |     |
| 1298 | 001127 | 000    |     |
| 1299 | 001130 | 000    |     |
| 1300 |        | 001132 |     |
| 1301 | 001132 | 030460 |     |
| 1302 | 001134 | 031462 |     |
| 1303 | 001136 | 032464 |     |
| 1304 | 001140 | 033466 |     |
| 1305 | 001142 | 034470 |     |
| 1306 | 001144 | 000006 |     |
| 1307 | 001152 | 000    |     |
| 1308 |        | 001154 |     |
| 1309 | 001154 | 000010 |     |
| 1310 | 001164 | 000100 |     |
| 1311 | 001264 | 000110 |     |
| 1312 | 001374 | 005015 | 000 |
| 1313 | 001377 | 134    | 000 |
| 1314 | 001401 | 060    | 000 |
| 1315 | 001403 | 007    | 000 |
| 1316 | 001405 | 055    | 000 |
| 1317 | 001407 | 040    |     |
| 1318 | 001410 | 000040 |     |
| 1319 | 001412 | 004476 | 000 |
| 1320 |        | 001416 |     |

```

UNITNO: .BYTE 0
TYPFLG: .BYTE 0
PSCNT: .BYTE 0
ASFLG: .BYTE 0
DIGTAB: "01
        "23
        "45
        "67
        "89
OOIGITS: .BLKB 6
         .BYTE 0
         .EVEN
DRVTBL: .BLKB 8.
SLVTBL: .BLKB 64.
INBUF: .BLKB 72.
CRLF: .ASCIZ <CR><LF>
BKSLSH: .ASCIZ '\ '
ECHO: .ASCIZ '0'
BELL: .ASCIZ <7>
DASH: .ASCIZ '-'
SPACE2: .ASCII ' '
SPACE: .ASCIZ ' '
ANGTAB: .ASCIZ '>'<HT>
        .EVEN

```

```

;UNIT FOUND INDICATOR
;CONTAINS PASS COUNT
;I/O = YES/NO.
;RESERVE SPACE FOR CONVERTED DIGITS
;TERMINATOR
;A 0/-1 = DRIVE NOT TO BE/TO BE TESTED
;A 0/-1 = SLAVE NOT TO BE/TO BE TESTED
;TELETYPE INPUT BUFFER
;MISCELLANEOUS ASCII CHARACTERS

```

SBTTL TIME SPECIFICATION TABLE

THE BELOW TABLE CONTAINS THE SPECIFIED FUNCTION TIMES IN TENS OF MICROSECONDS. NOTE THAT WHEN TIMES ARE TYPED THAT THEY ARE TYPED IN MICROSECONDS (BY APPENDING A 0).  
FORMAT IS

|      | WORD   | MAX    | MIN    | TIME IN MS | FUNCTION | TEST # |
|------|--------|--------|--------|------------|----------|--------|
| 1321 |        |        |        |            |          |        |
| 1322 |        |        |        |            |          |        |
| 1323 |        |        |        |            |          |        |
| 1324 |        |        |        |            |          |        |
| 1325 |        |        |        |            |          |        |
| 1326 |        |        |        |            |          |        |
| 1327 |        |        |        |            |          |        |
| 1328 | 001416 | 000000 | 000000 |            |          |        |
| 1329 | 001422 | 036050 | 035230 |            |          |        |
| 1330 | 001426 | 001666 | 001546 |            |          |        |
| 1331 | 001432 | 001522 | 001356 |            |          |        |
| 1332 | 001436 | 002506 | 001332 |            |          |        |
| 1333 | 001442 | 007164 | 006344 |            |          |        |
| 1334 | 001446 | 000500 | 000360 |            |          |        |
| 1335 | 001452 | 000632 | 000454 |            |          |        |
| 1336 | 001456 | 002506 | 001332 |            |          |        |
| 1337 | 001462 | 000500 | 000360 |            |          |        |
| 1338 | 001466 | 000562 | 000512 |            |          |        |
| 1339 | 001472 | 002506 | 001332 |            |          |        |
| 1340 | 001476 | 003206 | 002056 |            |          |        |
| 1341 | 001502 | 003206 | 002056 |            |          |        |
| 1342 | 001506 | 002412 | 001666 |            |          |        |
| 1343 | 001512 | 002234 | 001522 |            |          |        |
| 1344 | 001516 | 002626 | 002354 |            |          |        |
| 1345 | 001522 | 002506 | 002234 |            |          |        |
| 1346 | 001526 | 004540 | 004230 |            |          |        |
| 1347 | 001532 | 004716 | 004552 |            |          |        |
| 1348 | 001536 | 023564 | 023110 |            |          |        |
| 1349 | 001542 | 024240 | 023730 |            |          |        |
| 1350 | 001546 | 004336 | 004172 |            |          |        |
| 1351 | 001552 | 004336 | 004172 |            |          |        |
| 1352 |        |        |        |            |          |        |
| 1353 |        |        |        |            |          |        |

| STIMTBL | WORD | MAX   | MIN   | TIME IN MS  | FUNCTION        | TEST # |
|---------|------|-------|-------|-------------|-----------------|--------|
|         |      | 0     | 0     |             | SPARE           |        |
|         |      | 15400 | 15000 | 154.0-150.0 | WRITE FROM BOT  | TST001 |
|         |      | 00950 | 00870 | 9.5-8.7     | WRITE START     | TST002 |
|         |      | 00850 | 00750 | 8.9-8.5     | WRITE SHUTDOWN  | TST003 |
|         |      | 01350 | 00730 | 13.5-7.3    | WRITE STLDOWN   | TST004 |
|         |      | 03700 | 03300 | 37.0-33.0   | READ FROM BOT   | TST005 |
|         |      | 00320 | 00240 | 3.2-2.4     | READ START      | TST006 |
|         |      | 00410 | 00300 | 4.1-3.00    | READ SHUTDOWN   | TST007 |
|         |      | 01350 | 00730 | 13.5-7.3    | READ SETTLEDOWN | TST010 |
|         |      | 00320 | 00240 | 3.2-2.4     | RD REV START    | TST011 |
|         |      | 00370 | 00330 | 3.7-3.3     | RD REV SHUTDOWN | TST012 |
|         |      | 01350 | 00730 | 13.5-7.3    | RD REV STLDOWN  | TST013 |
|         |      | 01670 | 01070 | 16.7-10.7   | TRN RND DLY F-R | TST014 |
|         |      | 01670 | 01070 | 16.7-10.7   | TRN RND DLY R-F | TST015 |
|         |      | 01290 | 00950 | 12.9-9.5    | GAP SIZE STOP   | TST016 |
|         |      | 01180 | 00850 | 11.8-8.5    | GAP SIZE STRT   | TST017 |
|         |      | 01430 | 01260 | 14.3-12.6   | GAP SIZE INTER  | TST020 |
|         |      | 01350 | 01180 | 13.5-11.8   | GAP CONSISANCY  | TST021 |
|         |      | 02400 | 02200 | 24.0-22.0   | DAT TIME 800BPI | TST022 |
|         |      | 02510 | 02410 | 25.1-24.1   | DAT TIME 1600PE | TST023 |
|         |      | 10100 | 09800 | 101.0-98.0  | ERASE           | TST024 |
|         |      | 10400 | 10200 | 104.0-102.0 | WRT FILE MARK   | TST025 |
|         |      | 02270 | 02170 | 22.7-21.7   | TAPE SPEED FWD  | TST026 |
|         |      | 02270 | 02170 | 22.7-21.7   | TAPE SPEED REV  | TST027 |

NOTE: TEST 26 AND 27 REQUIRE PRERECORDED 800BPI SKEW TAPE.



SBTTL GAP TIME SPECIFICATION TABLE  
: THIS TABLE CONTAINS THE GAP SIZES (IN TENS OF MICROSECONDS) FOR EACH  
: OF THE 16 GAPS RECORDED BY THE GAP CONSISTANCY TEST (TST021).  
: NOTE: GAP #'S ARE IN OCTAL.

1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361 001556 002544 002354  
1362 001562 002652 002506  
1363 001566 002722 002506  
1364 001572 002710 002474  
1365 001576 002570 002260  
1366 001602 002556 002114  
1367 001606 002532 002114  
1368 001612 002532 002114  
1369 001616 002506 002176  
1370 001622 002474 002176  
1371 001626 002474 002176  
1372 001632 002474 002176  
1373 001636 002474 002176  
1374 001642 002474 002176  
1375 001646 002474 002176  
1376 001652 002474 002176

|          | .WORD | MAX,MIN(10)   | ; TIME IN MS(10) | GAP #  | DELAY IN MS(10) |
|----------|-------|---------------|------------------|--------|-----------------|
| GTIMTBL: | .WORD | 01380.,01260. | :13.8-12.6       | GAP-0  | 0 MS            |
|          | .WORD | 01450.,01350. | :14.5-13.5       | GAP-1  | 1.0 MS          |
|          | .WORD | 01490.,01350. | :14.9-13.5       | GAP-2  | 2.0 MS          |
|          | .WORD | 01480.,01340. | :14.8-13.4       | GAP-3  | 3.0 MS          |
|          | .WORD | 01400.,01200. | :14.0-12.0       | GAP-4  | 4.0 MS          |
|          | .WORD | 01390.,01100. | :13.9-11.0       | GAP-5  | 5.0 MS          |
|          | .WORD | 01370.,01100. | :13.7-11.0       | GAP-6  | 6.0 MS          |
|          | .WORD | 01370.,01100. | :13.7-11.0       | GAP-7  | 7.0 MS          |
|          | .WORD | 01350.,01150. | :13.5-11.5       | GAP-10 | 8.0 MS          |
|          | .WORD | 01340.,01150. | :13.4-11.5       | GAP-11 | 9.0 MS          |
|          | .WORD | 01340.,01150. | :13.4-11.5       | GAP-12 | 10.0 MS         |
|          | .WORD | 01340.,01150. | :13.4-11.5       | GAP-13 | 11.0 MS         |
|          | .WORD | 01340.,01150. | :13.4-11.5       | GAP-14 | 12.0 MS         |
|          | .WORD | 01340.,01150. | :13.4-11.5       | GAP-15 | 13.1 MS         |
|          | .WORD | 01340.,01150. | :13.4-11.5       | GAP-16 | 14.1 MS         |
|          | .WORD | 01340.,01150. | :13.4-11.5       | GAP-17 | 15.1 MS         |

1377  
1378  
1379 001656 000000  
1380 001660 014765  
1381 001662 015007  
1382 001664 015027  
1383 001666 015051  
1384 001670 015075  
1385 001672 015117  
1386 001674 015136  
1387 001676 015160  
1388 001700 015203  
1389 001702 015225  
1390 001704 015252  
1391 001706 015301  
1392 001710 015332  
1393 001712 015363  
1394 001714 015411  
1395 001716 015440  
1396 001720 015470  
1397 001722 015513  
1398 001724 015537  
1399 001726 015564  
1400 001730 015606  
1401 001732 015631  
1402 001734 015653  
1403  
1404  
1405 001736 007166  
1406 001740 007432  
1407 001742 007516  
1408 001744 007574  
1409 001746 007664  
1410 001750 007772  
1411 001752 010056  
1412 001754 010142  
1413 001756 010242  
1414 001760 010362  
1415 001762 010460  
1416 001764 010570  
1417 001766 010730  
1418 001770 011022  
1419 001772 011130  
1420 001774 011224  
1421 001776 011334  
1422 002000 011454  
1423 002002 011760  
1424 002004 012110  
1425 002006 012240  
1426 002010 012360  
1427 002012 012714  
1428 002014 013042

SBTTL TEST HEADER POINTERS  
:THE BELOW TABLE CONTAINS POINTERS TO EACH TEST'S DESCRIPTOR  
NAMPTR: .WORD 0

.WORD A.T001  
.WORD A.T002  
.WORD A.T003  
.WORD A.T004  
.WORD A.T005  
.WORD A.T006  
.WORD A.T007  
.WORD A.T010  
.WORD A.T011  
.WORD A.T012  
.WORD A.T013  
.WORD A.T014  
.WORD A.T015  
.WORD A.T016  
.WORD A.T017  
.WORD A.T020  
.WORD A.T021  
.WORD A.T022  
.WORD A.T023  
.WORD A.T024  
.WORD A.T025  
.WORD A.T026  
.WORD A.T027

:TABLE OF TEST STARTING ADDRESSES  
TSTTBL. .WORD TST000

.WORD TST001  
.WORD TST002  
.WORD TST003  
.WORD TST004  
.WORD TST005  
.WORD TST006  
.WORD TST007  
.WORD TST010  
.WORD TST011  
.WORD TST012  
.WORD TST013  
.WORD TST014  
.WORD TST015  
.WORD TST016  
.WORD TST017  
.WORD TST020  
.WORD TST021  
.WORD TST022  
.WORD TST023  
.WORD TST024  
.WORD TST025  
.WORD TST026  
.WORD TST027

```

1429 002016 000000 TIB: .WORD 0
1430 ;ROUTINE TO LOAD SSOFTWARE SWR
1431
1432 002020 022737 000176 001000 GTSWR: CMP #SWREG, SWR ;BRANCH IF SOFTWARE SWR
1433 002026 001027 BNE 2$ ;NOT INVOKED
1434 002030 004737 002354 JSR PC, .SAVE ;SAVE REGISTERS ON THE STACK
1435 002034 000004 015702 TYPE, L.SWR
1436 002040 017702 176734 MOV #SWR, R2
1437 002044 004737 002426 JSR PC, TYPOCT
1438 002050 000004 015711 TYPE, L.NEW
1439 002054 004737 003272 JSR PC, .INPUT ;GET USER INPUT
1440 002060 122737 000015 001264 CMPB #CR, #INBUF ;EXIT IF FIRST CHAR IS <CR>
1441 002066 001405 BEQ 1$
1442 002070 004737 003056 JSR PC, CNVTA0 ;CONERT ASCII TO OCTAL
1443 002074 013777 001116 176676 MOV #OCTALO, #SWR ;SET NEW SWITCH REG CONTENTS
1444 002102 004737 002376 1$: JSR PC, .RESTORE
1445 002106 000207 2$: RTS PC
1446
1447
1448 .SBTTL PROGRAM SUBROUTINES
1449 .SBTTL TYPE SUBROUTINE
1450 ;;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1451 ;;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1452 ;;CALL: TYPE ;A TRAP TYPE INSTRUCTION
1453 ;; MESAOR ;MESAOR IS FIRST ADDRESS OF ASCIZ STRING
1454 ;;
1455 ;;TAGS USED BY THE TYPE ROUTINE BELOW
1456 $HT=11 ;HORIZONTAL TAB
1457 002110 000 ;CONTAINS NULL CHARACTER
1458 002111 002 ;CONTAINS # OF FILLER CHARACTERS
1459 002112 000 ;CONTAINS TELEPRINTER AVAILABLE FLAG
1460 ;0/377 = AVAIL/NOT AVAIL
1461 002113 000 ;CONTAINS KEYBOARD AVAILABLE FLAG
1462 002114 177564 $TPS: .WORD 177564 ;ADDRESS OF TELEPRINTER STATUS REGISTER
1463 002116 177566 $TPB: .WORD 177566 ;ADDRESS OF TELEPRINTER DATA BUFFER
1464 002120 000 $CHARCNT: .BYTE 0 ;CONTAINS # OF CHARS TYPED
1465 002121 000 $CNTRL0: .BYTE 0 ;CONTAINS CONTROL 0 CHAR (IF TYPED)
1466 002122 005015 000 $CRLF: .ASCIZ <15><12>
1467 002126 002126 .EVEN
1468 002126 000000 RDSW: .WORD 0
1469
1470 002130 010046 .TYPE: MOV R0, -(SP) ;SAVE R0
1471 002132 017600 000002 MOV #2(SP), R0 ;GET MESSAGE ADDRESS
1472 002136 062766 000002 000002 ADD #2, 2(SP) ;ADJUST RETURN PC
1473 002144 105037 002121 CLR B $CNTRL0
1474
1475 002150 105737 002121 TYPE1: TSTB $CNTRL0 ;BRANCH IF CONTROL 0(10) WASN'T TYPED
1476 002154 001410 BEQ TYPE2
1477 002156 000004 002122 TCRLF: TYPE, $CRLF ;TYPE <CR><LF>
1478 002162 105737 002126 TSTB RDSW
1479 002166 100006 BPL TYPE3
1480 002170 005037 002126 CLR RDSW
1481 002174 000207 RTS PC
1482 002176 112046 TYPE2: MOV B (R0)+, -(SP) ;PUSH CHARACTER TO BE TYPED ONTO STACK
1483 002200 001003 BNE TYPE4 ;BRANCH IF NOT THE TERMINATOR
1484 002202 005726 TST (SP)+ ;POP TERMINATOR CHAR OFF THE STACK

```

```

1485 002204 012600 TYPE3: MOV (SP)+,R0 ;;RESTORE R0
1486 002206 000002 RTI ;;RETURN TO CALLER
1487
1488 002210 122716 000011 TYPE4: CMPB #SHT,(SP) ;;BRANCH IF HORIZONTAL TAB <HT>
1489 002214 001445 BEQ 9$
1490 002216 004737 002250 JSR PC,5$ ;;TYPE CHARACTER
1491 002222 122726 000012 3$: CMPB #12,(SP)+ ;;CHECK IF CHARACTER WAS A LINE FEED
1492 002226 001350 BNE TYPE1 ;;BRANCH IF NOT LINE FEED
1493 002230 013746 002110 MOV $NULL,-(SP) ;;GET # OF FILLERS REQUIRED AND FILLER
1494 ;;CHARACTER.
1495
1496 002234 105366 000001 4$: DECB 1(SP) ;;DECREMENT FILLERS REQ. COUNT
1497 00 240 002770 BLT 3$ ;;BRANCH IF NO MORE FILLERS ARE REQUIRED
1498 00 242 004737 002250 JSR PC,5$ ;;TYPE FILLER CHARACTER
1499 002246 000772 BR 4$
1500
1501 002250 105777 177640 5$: TSTB @STPS ;;WAIT FOR OUTPUT DEVICE
1502 002254 100375 BPL .-4
1503 002256 122737 000017 002121 CMPB #17,@$CNTRL0 ;;CHECK IF CONTROL 0 WAS TYPED
1504 002264 001403 BEQ 6$ ;;STOP TYPING MESSAGE IF 10 WAS TYPED
1505 002266 116677 000002 177622 MOVB 2(SP),@STPB ;;OUTPUT CHARACTER
1506 002274 122766 000015 000002 6$: CMPB #15,2(SP) ;;BRANCH IF NOT <CR>
1507 002302 001003 BNE 7$
1508 002304 105037 002120 CLRB $CHARCNT ;;CLEAR CHARACTERS TYPED COUNT
1509 002310 000406 BR 8$
1510 002312 122766 000012 000002 7$: CMPB #12,2(SP) ;;BRANCH IF <LF> OR 'NULL'
1511 002320 002002 BGE 8$
1512 002322 105237 002120 INCB $CHARCNT ;;INCREMENT CHARACTER TYPED COUNT
1513 002326 000207 8$: RTS PC
1514
1515 ;;HORIZONTAL TAB <HT> PROCESSER
1516 002330 112716 000040 9$: MOVB #40,(SP) ;;LOAD 'SPACE'
1517 002334 004737 002250 10$: JSR PC,5$ ;;TYPE 'SPACE'
1518 002340 132737 000007 002120 BITB #7,$CHARCNT ;;TYPE SPACES UNTIL A MULTIPLE
1519 002346 001372 BNE 10$ ;;OF 8 CHARACTERS HAVE BEEN TYPED
1520 002350 105726 TSTB (SP)+ ;;POP SPACE
1521 002352 000676 BR TYPE1 ;;GET NEXT CHARACTER
1522
1523 ;SUBROUTINE TO SAVE GENERAL REGISTERS ON THE STACK
1524 ;CALL: SAVE
1525 002354 010546 .SAVE: MOV R5,-(SP) ;SAVE REGISTERS ON THE STACK
1526 002356 010446 MOV R4,-(SP)
1527 002360 010346 MOV R3,-(SP)
1528 002362 010246 MOV R2,-(SP)
1529 002364 010146 MOV R1,-(SP)
1530 002366 010046 MOV R0,-(SP)
1531 002370 016646 000014 MOV 14(SP),-(SP) ;GET RETURN PC
1532 002374 000207 RTS PC ;RETURN
1533
1534 ;SUBROUTINE TO RESTORE GENERAL REGISTERS FROM THE STACK
1535 ;CALL: RESTORE
1536 002376 012666 .RESTORE:MOV (SP)+,14(SP) ;MOVE RETURN PC
1537 002402 012600 MOV (SP)+,R0 ;RESTORE REGISTERS
1538 002404 012601 MOV (SP)+,R1
1539 002406 012602 MOV (SP)+,R2
1540 002410 012603 MOV (SP)+,R3
    
```

```

1541 002412 012604      MOV      (SP)+,R4
1542 002414 012605      MOV      (SP)+,R5
1543 002416 000207      RTS      PC      ;RETURN
1544
1545      ;SUBROUTINE TO CONVERT OCTAL DATA TO ASCII
1546      ;CALL:  MOV      NUMBER,R2      ;MOVE NUMBER TO R2
1547      ;      JSR      PC,CNV OCT
1548
1549 002420 110637 001126  CNVOCT: MOVB    SP,TYPFLG      ;SET DO NOT TYPE FLAG
1550 002424 000402      BR      CNVTO
1551
1552      .SBTTL      OCTAL TO ASCII & TYPE ROUTINE
1553      ;SUBROUTINE TO CONVERT OCTAL NUMBER TO ASCII AND TYPE IT OUT
1554      ;CALL:  MOV      NUMBER,R2      ;PUT # IN R2
1555      ;      JSR      PC,TYP OCT      ;CALL ROUTINE
1556
1557 002426 105037 001126  TYP OCT: CLRB   @#TYPFLG      ;SET TYPE FLAG
1558 002432 000402      CNVTO:
1559 002432 004737 002354      JSR      PC,SAVE      ;SAVE REGISTERS ON THE STACK
1560 002436 012704 001144      MOV      #00DIGITS,R4      ;SET PTR TO OUTPUT
1561 002442 005003      CLR      R3      ;R3 WILL CONTAIN OCTAL DIGIT
1562 002444 010201      MOV      R2,R1      ;GET # TO BE TYPED
1563 002446 006302 15:      ASL      R2      ;SHIFT #
1564 002450 006103      ROL      R3
1565 002452 012700 000006      MOV      #6,R0      ;SET DIGIT COUNTER
1566 002456 000404      BR      3$
1567
1568 002460 006302 25:      ASL      R2      ;SHIFT # 3 PLACES LEFT
1569 002462 006103      ROL      R3
1570 002464 005301      DEC      R1
1571 002466 001374      BNE     2$
1572 002470 012701 000003 35:      MOV      #3,R1      ;SET SHIFT COUNTER
1573 002474 116324 001132      MOVB    DIGTAB(R3),(R4)+      ;MOVE ASCII EQUIV TO OUTPUT
1574 002500 005003      CLR      R3
1575 002502 005300      DEC      R0      ;DECREMENT DIGIT COUNT
1576 002504 001365      BNE     2$      ;GET NEXT DIGIT
1577 002506 105737 001126      TSTB    @#TYPFLG      ;BRANCH IF ASCII IS
1578 002512 001002      BNE     4$      ;NOT TO BE TYPED
1579 002514 000004 001144      TYPE,00DIGITS
1580 002520 000004 45:
1581 002520 004737 002376      JSR      PC,.RESTORE      ;RESTORE REGISTERS FROM THE STACK
1582 002524 000207      RTS      PC
1583
1584
1585      ;SUBROUTINE TO CONVERT OCTAL DATA TO DECIMAL ASCII
1586      ;CALL:  MOV      NUMBER,R2      ;MOVE NUMBER TO R2
1587      ;      JSR      PC,CNV DEC
1588
1589 002526 110637 001126  CNV DEC: MOVB    SP,@#TYPFLG      ;SET DO NOT TYPE FLAG
1590 002532 000402      BR      CNVTD
1591
1592      .SBTTL      OCTAL TO DECIMAL & TYPE ROUTINE
1593      ;THIS ROUTINE CONVERTS AN OCTAL # TO DECIMAL ASCII AND TYPES IT OUT
1594      ;CALL:  MOV      NUMBER,R2      ;PUT # IN R2
1595      ;      JSR      PC,TYP DEC      ;CALL ROUTINE
1596 002534 105037 001126  TYP DEC: CLRB   @#TYPFLG      ;SET TYPE FLAG
    
```

```

1597 002540
1598 002540 004737 002354
1599 002544 005000
1600 002546 012704 001144
1601 002552 005003
1602 002554 166002 002634
1603 002560 103402
1604 002562 005203
1605 002564 000773
1606 002566 066002 002634
1607 002572 116324 001132
1608 002576 062700 000002
1609 002602 005760 002634
1610 002606 001361
1611 002610 112724 000060
1612 002614 105737 001126
1613 002620 001002
1614 002622 000004 001144
1615 002626
1616 002626 004737 002376
1617 002632 000207
1618
1619 002634 023420
1620 002636 001750
1621 002640 000144
1622 002642 000012
1623 002644 000001
1624 002646 000000
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637 002650 010246
1638 002652 010346
1639 002654 006302
1640 002656 006302
1641 002660 010203
1642 002662 000004 014745
1643 002666 016302 001416
1644 002672 004737 002534
1645 002676 000004 001405
1646 002702 016302 001420
1647 002706 004737 002534
1648 002712 000004 001412
1649 002716 000004 014755
1650 002722 013702 001012
1651 002726 004737 002534
1652 002732 000004 001374
    
```

```

CMVTD:
JSR PC, .SAVE ; SAVE REGISTERS ON THE STACK
CLR R0 ; R0 IS INDEX TO DECIMAL CONSTANT
MOV #00IGITS, R4 ; SET OUTPUT PTR
15: CLR R3 ; R3 CONTAINS DECIMAL DIGIT
25: SUB DCONST(R0), R2 ; SUBTRACT DECIMAL CONSTANT UNTIL
; INPUT # GOES NEGATIVE
; KEEPING TRACK OF SUBTRACTIONS
BLO 35
INC R3
BR 25
35: ADD DCONST(R0), R2 ; ADD BACK CONSTANT WHEN NEGATIVE
MOVB DIGTAB(R3), (R4)+ ; MOVE ASCII EQUIVALENT
ADD #2, R0 ; NEXT CONSTANT
TST DCONST(R0) ; UNTIL ALL CONSTANTS DONE
BNE 15
MOVB #0, (R4)+ ; LAST DIGIT IS 0
TSTB @#TYPFLG ; BRANCH IF ASCII IS
BNE 45 ; NOT TO BE TYPED
45: JSR PC, .RESTORE ; RESTORE REGISTERS FROM THE STACK
RTS PC

DCONST: .WORD 10000.
; .WORD 1000.
; .WORD 100.
; .WORD 10.
; .WORD 1.
; .WORD 0 ; TERMINATOR

.SBTTL TYPE SPECIFIED TIMES ROUTINE
; THIS SUBROUTINE OUTPUTS THE TIME SPECIFICATIONS FOR THE TEST
; AND ALSO THE ACTUAL TIME RECORDED (ATIME)
; FORMAT OF LINE TYPED
; RANGE=(AAAAA-BBBBBB) ACTUAL=CCCCC
; WHERE: AAAAAA IS MAXIMUM TIME FOR TEST (STIMTBL(TSTNUMX4)).
; BBBB BB IS MINIMUM TIME FOR TEST (STIMTBL(TSTNUMX4+2)).
; CCCCCC IS ACTUAL TIME RECORDED BY TEST (ATIME).
CALL: MOVB TEST NUMBER, R2 ; LOAD TEST NUMBER
MOV #ATIME, @#ATIME ; MOVE TIME TO ATIME
JSR PC, OUTSPC ; SAVE R2 & R3 ON THE STACK
OUTSPC: MOV R2, -(SP)
MOV R3, -(SP)
ASL R2 ; MULTIPLY TEST # TIMES 4
ASL R2 ; TO FORM INDEX INTO STIMTBL
MOV R2, R3 ; R3 CONTAINS INDEX INTO TABLE
TYPE, L.RNG
MOV STIMTBL(R3), R2 ; GET MAXIMUM SPEC TIME
JSR PC, TYPDEC ; CONVERT TO DECIMAL & TYPE
TYPE, DASH
MOV STIMTBL+2(R3), R2 ; GET MINIMUM TIME
JSR PC, TYPDEC ; CONVERT TO DECIMAL & TYPE
TYPE, ANGTAB
TYPE, L.ACT
MOV @#ATIME, R2 ; GET ACTUAL TIME
JSR PC, TYPDEC ; CONVERT TO DECIMAL & TYPE
TYPE, CRLF
    
```

```

1653 002736 012603
1654 002740 012602
1655 002742 000207
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665 002744 010246
1666 002746 010346
1667 002750 113703 001120
1668 002754 006303
1669 002756 006303
1670 002760 000004 014745
1671 002764 016302 001556
1672 002770 004737 002534
1673 002774 000004 001405
1674 003000 016302 001560
1675 003004 004737 002534
1676 003010 000004 001412
1677 003014 000004 014755
1678 003020 013702 001012
1679 003024 004737 002534
1680 003030 000004 014434
1681 003034 113702 001120
1682 003040 004737 002426
1683 003044 000004 001374
1684 003050 012603
1685 003052 012602
1686 003054 000207
1687
1688
1689
1690
1691 003056
1692 003056 004737 002354
1693 003062 012700 001264
1694 003066 012701 001116
1695 003072 005011
1696 003074 005061 000002
1697 003100 122710 000015
1698 003104 001414
1699 003106 112002
1700 003110 042702 177770
1701 003114 012703 000003
1702 003120 006311
1703 003122 006161 000002
1704 003126 005303
1705 003130 001373
1706 003132 050211
1707 003134 000761
1708 003136
    
```

```

MOV (SP)+,R3
MOV (SP)+,R2
RTS PC ;RETURN

;SRTL TYPE GAP TIMES SUBROUTINE
;THIS SUBROUTINE IS USED TO TYPE THE SPECIFIED GAP SIZES (RECORDED IN
;TST021). IT IS CALLED BY THE GAPOK ROUTINE IF THE GAP SIZE IS OUT OF
;RANGE VIA THE HLT ROUTINE (HLT+2).
CALL: MOVB #GAP,GAP ;LOAD GAP # INTO GAP
MOV #TIME,ATIME ;LOAD ACTUAL TIME INTO ATIME
JSR PC,OUTGAP

OUTGAP: MOV R2,-(SP) ;SAVE R2 AND R3
MOV R3,-(SP)
MOVB GAP,R3 ;GET GAP #
ASL R3
ASL R3
TYPE,L,RNG
MOV #GTIME,R3 ;GET MAX TIME
JSR PC,TYPDEC ;CONVERT TO DECIMAL & TYPE
TYPE,DASH
MOV #GTIME+2,R3 ;GET MIN TIME
JSR PC,TYPDEC ;CONVERT TO DECIMAL & TYPE
TYPE,ANGTAB
TYPE,L,ACT
MOV #ATIME,R2 ;GET ACTUAL TIME
JSR PC,TYPDEC ;CONVERT TO DECIMAL & TYPE
TYPE,E,GAP
MOVB #GAP,R2 ;GET GAP #
JSR PC,TYPDEC ;TYPE GAP #
TYPE,CRLF
MOV (SP)+,R3 ;RESTORE R3 AND R2
MOV (SP)+,R2
RTS PC

;SRTL ASCII TO OCTAL CONVERT SUBROUTINE
;SUBROUTINE TO CONVERT ASCII DATA TO OCTAL. CONVERTED OCTAL DATA
;IS LEFT IN OCTALO <15-00>.
CNVTAO: JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
MOV #INBUF,R0 ;SET PTR TO ASCII DATA
MOV #OCTALO,R1 ;GET ADDRESS OF OCTAL DATA
CLR (R1) ;CLEAR OUT OLD OCTAL DATA
CLR 2(R1)
1$: CMPB #CR,(R0) ;<CR> TERMINATES INPUT
BEQ 3$
MOVB (R0)+,R2 ;GET 'OCTAL' DATA
BIC #177770,R2 ;STRIP UNUSED BITS
MOV #3,R3 ;SET SHIFT COUNT
2$: ASL (R1) ;SHIFT LAST
ROL 2(R1) ;OCTAL DIGIT
DEC R3
BNE 2$
BIS R2,(R1) ;AND INSERT THIS DIGIT
BR 1$ ;GO GET NEXT DIGIT
3$:
    
```

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 42  
 DZTEEA.P11 31-MAR-77 00:00 ASCII TO OCTAL CONVERT SUBROUTINE

```

1709 003136 004737 002376      JSR   PC, .RESTORE      ;RESTORE REGISTERS FROM THE STACK
1710 003142 000207              RTS    PC                ;RETURN
1711
1712      .SBTTL      PUBLISH SUBROUTINE
1713      ;THE PUBLISH SUBROUTINE AVERAGES THE RECORDED TIMES FOR EACH TEST IT-
1714      ;ERATION (IF 16. ITERATIONS) AND PLACES THE AVERAGE RESULT IN 'ATIME'.
1715      ;IT TYPES THE NAME OF THE FUNCTION THAT WAS TIMED, THE TIME SPEC-
1716      ;IFICATION AND THE ACTUAL TIME .
1717
1718 003144
1719 003144 004737 002354      JSR   PC, .SAVE         ;SAVE REGISTERS ON THE STACK
1720 003150 012700 001014      MOV   @ATIMTBL, R0     ;GET TABLE ADDRESS CONTAINING TIMES
1721 003154 113701 001121      MOVB  @ITCNT, R1       ;GET # OF ENTRIES (GIVEN BY ITERATION COUNT)
1722 003160 122701 000001      CMPB  #1, R1           ;BRANCH IF SINGLE ITERATION
1723 003164 001423              BEQ   #4$              ;
1724 003166 005002              CLR   R2               ;CLEAR 'SUM' REGISTERS
1725 003170 005003              CLR   R3               ;
1726 003172 122701 000020      CMPB  #16., R1         ;BRANCH IF 16. ITERATIONS
1727 003176 001402              BEQ   #1$              ;
1728 003200 000000              HALT                    ;ITERATION COUNT MUST BE 1 OR 16.
1729 003202 000777              BR    .                ;DO NOT CHANGE POSIT OF SW11
1730                                     ;WHEN TEST IS RUNNING.
1731
1732 003204 062002      1$:   ADD   (R0)+, R2     ;SUM INDIVIDUAL TIMES
1733 003206 005503              ADC   R3
1734 003210 005301              DEC   R1
1735 003212 001374              BNE   #1$
1736
1737 003214 012700 000004      2$:   MOV   #4, R0
1738 003220 006203      3$:   ASR   R3             ;SHIFT TIME IN R3 & R2 4 PLACES
1739 003222 006002              ROR   R2             ;RIGHT = DIVIDE BY 16.
1740 003224 005300              DEC   R0
1741 003226 001374              BNE   #3$
1742 003230 010237 001012      MOV   R2, @ATIME      ;MOVE AVERAGED TIMES
1743
1744 003234 113700 001122      4$:   MOVB  @TSTNUM, R0   ;GET TEST #
1745 003240 006300              ASL   R0
1746 003242 016037 001656 003252  MOV   NAMPTR(R0), #5$ ;GET TEST NAME STRING ADDRESS
1747 003250 000004              TYPE
1748 003252 000000      5$:   .WORD  0
1749 003254 113702 001122      MOVB  @TSTNUM, R2     ;GET TEST #
1750 003260 004737 002650      JSR   PC, OUTSPC      ;OUTPUT TIMES
1751 003264 004737 002376      JSR   PC, .RESTORE   ;RESTORE REGISTERS FROM THE STACK
1752 003270 000207              RTS    PC
1753
1754      .SBTTL      INPUT SUBROUTINE
1755      ;SUBROUTINE TO GET TTY INPUT
1756      ;CALL: JSR   PC, .INPUT
1757      ;INPUT DATA IS RETURNED IN BUFFER BEGINNING AT INBUF.
1758
1759 003272 010046      .INPUT: MOV   R0, -(SP)   ;SAVE R0 ON THE STACK
1760 003274 012700 001264      1$:   MOV   #INBUF, R0
1761 003300 105737 177560      2$:   TSTB  @TKS
1762 003304 100375              BPL   #2$
1763
1764 003306 113746 177562              MOVB  @TKB, -(SP)    ;GET CHARACTER

```





```

1816          .SBTTL          ERROR SERVICE ROUTINES
1817          ;ROUTINE TO PROCESS ERROR TRAPS (TRAPS TO 4)
1818 003542 000000          ERTRP: HALT
1819
1820          ;ERROR SERVICE ROUTINE
1821          ;THIS ROUTINE PROCESSES TWO TYPES OF ERRORS (OUT OF RANGE AND HARDWARE)
1822          ;THE CALLS FOR AN OUT OF RANGE ERROR ARE <HLT+1>, <HLT+2> AND, FOR A
1823          ;HARDWARE ERROR THE CALL IS <HLT>.
1824
1825 003544 004737 002354          .HLT: JSR PC, SAVE          ;SAVE REGISTERS ON THE STACK
1826 003550 110637 001123          1$:  MOVB SP, @ERFLG          ;SET ERROR FLAG
1827 003554 032777 020000 175216  BIT #SW13, @SWR          ;BRANCH IF NO TYP0UT
1828 003562 001075          BNE 4$
1829 003564 000004 014235          TYPE, E.HDR
1830 003570 113702 001122          MOVB @TSTNUM, R2          ;GET TEST #
1831 003574 004737 002426          JSR PC, TYP0CT          ;AND TYPE IT
1832 003600 016600 000016          MOV 16(SP), R0          ;GET RETURN PC
1833 003604 162700 000002          SUB #2, R0          ;NOW PC OF HLT CALL
1834 003610 111000          MOVB (R0), R0          ;NOW HLT CALL ITSELF
1835 003612 001417          BEQ 2$          ;BRANCH IF HLT
1836 003614 000004 014320          TYPE, E.HDR2
1837 003620 122700 000002          CMPEB #2, R0          ;BRANCH IF NOT HLT+2
1838 003624 001005          BNE 10$
1839 003626 004737 002744          JSR PC, OUTGAP          ;TYPE GAP SPECIFIED TIMES
1840 003632 000004 001374          TYPE, CRLF
1841 003636 000447          BR 4$
1842 003640 004737 002650          10$: JSR PC, OUTSPC          ;TYPE SPECIFIED TIMES
1843 003644 000004 001374          TYPE, CRLF
1844 003650 000442          BR 4$
1845 003652 016500 000014          2$:  MOV ER(R5), R0
1846 003656 032765 002300 000032  BIT #PE1600, TC(R5)
1847 003664 001403          BEQ 20$
1848 003666 042700 102100          RIC #102100, R0
1849 003672 000402          BR 21$
1850 003674 042700 102300          20$: BIC #102300, R0
1851 003700 005700          21$: TST R0
1852 003702 001003          BNE 22$
1853 003704 000004 014211          TYPE, E.SFT          ;TYPE SOFT ERROR MESSAGE
1854 003710 000434          BR 6$
1855
1856 003712 000004 014245          22$: TYPE, E.HDR1
1857 003716 010500          MOV R5, R0          ;GET FIRST ADDRESS OF REGS.
1858 003720 012701 000007          MOV #7, R1          ;TYPE FIRST 7 REGS.
1859 003724 012002          3$:  MOV (R0)+, R2          ;GET REG CONTENTS
1860 003726 004737 002426          JSR PC, TYP0CT          ;AND TYPE IT
1861 003732 000004 001407          TYPE, SPACE2
1862 003736 005301          DEC R1
1863 003740 001371          BNE 3$
1864 003742 016502 000032          MOV TC(R5), R2          ;GET CONTENTS OF TC REGISTER
1865 003746 004737 002426          JSR PC, TYP0CT
1866 003752 000004 001374          TYPE, CRLF
1867
1868 003756 032777 001000 175014  4$:  BIT #SW09, @SWR          ;BRANCH IF NO RING THE BELL
1869 003764 001402          BEQ 5$
1870 003766 000004 001403          TYPE, BELL
1871 003772 005777 175002          5$:  TST @SWR          ;HALT ON ERROR?
    
```

H04

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER MACY11 30(1046) 27-JUN-77 17:03 PAGE 45  
DZTEEA.P11 31-MAR-77 00:00 ERROR SERVICE ROUTINES

|      |        |        |        |      |              |                                   |
|------|--------|--------|--------|------|--------------|-----------------------------------|
| 1872 | 003776 | 100001 |        | BPL  | 6S           |                                   |
| 1873 | 004000 | 000000 |        | HALT |              |                                   |
| 1874 | 004002 |        | 6S:    |      |              |                                   |
| 1875 | 004002 | 004737 | 002376 | JSR  | PC, .RESTORE | ;RESTORE REGISTERS FROM THE STACK |
| 1876 | 004006 | 000002 |        | RTI  |              | ;RETURN                           |
| 1877 |        |        |        |      |              |                                   |
| 1878 |        |        |        |      |              |                                   |

```

1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895 004010 013705 001010
1896 004014 032777 000400 174756
1897 004022 001404
1898 004024 113702 001122
1899 004030 004737 002650
1900 004034 032777 040000 174736
1901 004042 001432
1902 004044 017701 174730
1903 004050 042701 177740
1904 004054 001406
1905 004056 020137 001122
1906 004062 001403
1907 004064 012737 007166 001002
1908 004072 004737 004634
1909 004076 004737 005070
1910 004102 105037 001123
1911 004106 013716 001002
1912 004112 010501
1913 004114 062701 000012
1914 004120 010500
1915 004122 062700 000006
1916 004126 000002
1917
1918 004130 105737 001123
1919 004134 001006
1920 004136 113700 001121
1921 004142 006300
1922 004144 013760 001012 001014
1923 004152 105237 001121
1924 004156 105737 001127
1925 004162 001410
1926 004164 032777 004000 174606
1927 004172 001004
1928 004174 122737 000020 001121
1929 004202 001320
1930 004204 032777 000037 174566
1931 004212 001002
1932 004214 011637 001002
1933 004220 032777 002000 174552
1934 004226 001005

```

```

.SBTTL SCOPE SUBROUTINE
SCOPE ROUTINE
THIS ROUTINE IS ENTERED UPON COMPLETION OF EACH SUBTEST
THE SCOPE ROUTINE:
OUTPUTS TIME SPEC ON EACH ITERATION IF SW08 IS SET
REPEATS TEST IF SW14 IS SET
STORES ACTUAL TIME FOR FUNCTION IN TIME TABLE (ATIMTBL)
PUBLISHES TIME IF SW10=0
UPDATES ITERATION COUNT AND IF ITERATIONS COMPLETE CONTINUES
TO NEXT TEST, OTHERWISE REPEATS TEST.
DELAYS BEFORE CONTINUING OR REPEATING TEST.
INITIALIZES DRIVE
RETURNS: RS=BASE ADDRESS OF TMO3 REGISTERS (ADDRESS OF CS1)
RI='DS' REG ADDRESS
RO='FC' REG ADDRESS

.SCOPE: MOV @TMBASE,RS ;SET RS TO FIRST TM REG
BIT @SW08,@SWR ;BRANCH IF SPECIFICATION LINE
BEQ 10$ ;NOT DESIRED ON EACH ITERATION
MOVB @TSTNUM,R2 ;GET TEST NUMBER
JSR PC,OUTSPC ;OUTPUT TIME RECORDED
10$: BIT @SW14,@SWR ;BRANCH IF CONTINUOUS LOOP
BEQ 2$ ;NOT DESIRED
1$: MOV @SWR,R1 ;GET SWITCHES
BIC @177740,R1 ;CLEAR ALL BUT TEST #
BEQ 11$ ;BRANCH IF ALL SELECTED
CMP R1,@TSTNUM ;BRANCH IF RUNNING SELECTED TEST
BEQ 11$
MOV @TST000,SCPADR ;RESTART AT TST000
11$: JSR PC,DELAY ;DELAY 350 MS
JSR PC,RHINIT ;INIT
CLRB @ERFLG ;CLEAR ERROR FLAG
MOV SCPADR,(SP)
MOV RS,R1 ;ADDRESS OF 'DS' REG IS IN R1
ADD @DS,R1
MOV RS,RO ;ADDRESS OF 'FC' REG IS IN RO
ADD @FC,RO
RTI

2$: TSTB @ERFLG ;BRANCH IF ERROR FLAG IS SET
BNE 3$
MOVB @ITCNT,RO ;GET ITERATION COUNT
ASL RO ;STORE TIME IN TABLE
MOV @ATIME,ATIMTBL(RO)
3$: INCB @ITCNT ;INCREMENT ITERATION COUNT
TSTB @PSCNT ;INHIBIT ITERATIONS ON
BEQ 4$ ;ON FIRST PASS
BIT @SW11,@SWR ;BRANCH IF SINGLE ITERATION DESIRED
BNE 4$
CMPB @16,@ITCNT ;BRANCH IF ITERATIONS INCOMPLETE
BNE 1$
4$: BIT @37,@SWR ;IF TEST SELECTED IS TEST 0
BNE 42$ ;TREAT AS ALL TESTS
MOV (SP),@SCPADR ;SET SCOPE ADDRESS TO NEXT TEST
40$: BIT @SW10,@SWR ;BRANCH IF NO PUBLICATION DESIRED
42$: BNE 5$
5$:

```

```

1935 004230 005737 005716          TST      CHNFLG          ;BRANCH IF IN CHAIN MODE
1936 004234 001002                   BNE      5$
1937 004236 004737 003144          JSR      PC PUBLISH    ;GO PUBLISH TEST DATA
1938 004242 105037 001121          CLR      001TCNT      ;RESET ITERATION COUNT
1939 004246 000676                   BR       1$
1940
1941          .SBTTL  TIMER SUBROUTINES
1942
1943          ;SUBROUTINE TO SYNCHRONIZE THE TIMER AND TURN IT ON.
1944          ;REGISTER 4 IS CLEARED, AND THE OSCILLATOR POLARITY IS MONITORED
1945          ;THE ROUTINE IS EXITED WHEN THE OSCILLATOR POLARITY CHANGES WITH R3
1946          ;SET TO INDICATE THE POLARITY OF THE OSCILLATOR.
1947          ;CALL:  JSR      PC, 1:MON
1948          ;RETURNS: R3 SET TO INDICATE LAST POLARITY (+24/-24=0/1)
1949                   R4 = 0
1950
1951 004250 005004          TIMON:  CLR      R4          ;CLEAR TIME COUNT
1952 004252 012703 000024          MOV      #24,R3        ;SET POLARITY TO '0' STATE
1953 004256 032765 000100 000024          BIT      #0SC,MR(R5)   ;BRANCH IF POLARITY IS '0'
1954 004264 001405                   BEQ      2$
1955 004266 032765 000100 000024          1$:    BIT      #0SC,MR(R5) ;WAIT FOR OSCILLATOR TO RETURN
1956 004274 001374                   BNE      1$
1957 004276 000405                   BR       4$
1958
1959 004300 005403          2$:    NEG      R3          ;NEGATE PREV POLARITY INDICATOR
1960 004302 032765 000100 000024          3$:    BIT      #0SC,MR(R5) ;WAIT FOR OSCILLATOR TO RETURN
1961 004310 001774                   BEQ      3$             ;TO '1' STATE
1962 004312 000207          4$:    RTS      PC
1963
1964          ;SUBROUTINE TO COUNT TIME
1965          ;EACH TIME THE OSCILLATOR TOGGLES (BIT <06> IN MR REG) REGISTER
1966          ;R4 IS INCREMENTED, AND THE REGISTER R3 IS NEGATED TO INDICATE
1967          ;THE LAST STATE OF THE OSCILLATOR.
1968          ;CALL  JMP      TIMER(R3)          ;R3 IS SET BY TIMON ROUTINE
1969          ;R2=RETURN ADDRESS TO CALLER
1970          ;NOTE: THE TIME TO EXECUTE THIS ROUTINE IS VERY CRITICAL. IT MUST BE
1971          ;LESS THAN 40 US.
1972
1973          ;ENTER HERE VIA JMP  TIMER(R3) WHEN R3=-24 (PREV STATE=1)
1974 004314 032765 000100 000024          TIMER1: BIT      #0SC,MR(R5) ;BRANCH IF CURRENT STATE IS '0'
1975 004322 001406                   BEQ      TIMER        ;GO INCREMENT TIME
1976 004324 000112                   JMP      (R2)          ;RETURN TO TEST
1977
1978          .=TIMER1+24
1979 004340 005403          TIMER:  NEG      R3          ;NEGATE PREV STATE INDICATOR
1980 004342 005204                   INC      R4          ;INCREMENT 'TICK' COUNT
1981 004344 100401                   BMI      TIMERR       ;BRANCH ON OVERFLOW
1982 004346 000112                   JMP      (R2)          ;RETURN TO TEST
1983 004350 000004 014346          TIMERR: TYPE,E.TIMOV    ;TYPE 'TIMER OVERFLOWED'
1984 004354 104400                   HLT                          ;REPORT HARDWARE ERROR
1985 004356 000177 174420          JMP      0JSCPADR      ;RETURN TO BEGINNING OF TEST
1986
1987          .=TIMER+24
1988          ;ENTER HERE VIA JMP  TIMER(R3) WHEN R3=+24 (PREV STATE=0)
1989 004364 032765 000100 000024          TIMER0: BIT      #0SC,MR(R5) ;BRANCH IF CURRENT STATE = '1'
1990 004372 001362                   BNE      TIMER
  
```

```

1991 004374 000112          JMP      (R2)
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 004376 004737 002354      TIMOK:   JSR      PC, .SAVE          ;SAVE REGISTERS ON THE STACK
2002 004376 004737 002354      MOV      #56, R0          ;GET TIME PER TICK
2003 004402 012700 000070      MOV      R4, R1          ;GET TICKS COUNT
2004 004406 010401
2005 004410 005002          CLR      R2              ;CLEAR SUMMING REGISTERS
2006 004412 005003          CLR      R3
2007 004414 060002      1$:     ADD      R0, R2          ;MULTIPLY TIME PER TICK
2008 004416 005503          ADC      R3              ;BY TICK COUNT
2009 004420 005301          DEC      R1
2010 004422 001374          BNE      1$
2011 004424 010246          MOV      R2, -(SP)       ;DIVIDE COUNT BY 10.
2012
2013 004426 010346          MOV      R3, -(SP)
2014 004430 012746 000012      MOV      #10, -(SP)
2015 004434 004737 004722      JSR      PC, DIVIDE
2016 004440 005726          TST      (SP)+
2017 004442 012637 001012      MOV      (SP)+, @#ATIME  ;DISCARD REMAINDER
2018 004446 113700 001122      MOV      @#TSTNUM, R0    ;STORE QUOTIENT
2019 004452 006300          ASL      R0              ;GET TEST #
2020 004454 006300          ASL      R0
2021 004456 023760 001012 001416  CMP      @#ATIME, STIMTBL(R0) ;CHECK THAT TIME IS WITHIN
2022 004464 101004          BHI      2$              ;LIMITS SPECIFIED
2023 004466 023760 001012 001420  CMP      @#ATIME, STIMTBL+2(R0)
2024 004474 101001          BHI      3$
2025 004476 104401      2$:     HLT+1              ;CALL ERROR ROUTINE
2026 004500      3$:
2027 004500 004737 002376      JSR      PC, .RESTORE    ;RESTORE REGISTERS FROM THE STACK
2028 004504 000207          RTS      PC              ;RETURN
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039 004506 004737 002354      GAPOK:   JSR      PC, .SAVE          ;SAVE REGISTERS ON THE STACK
2040 004506 004737 002354      MOV      #56, R0          ;GET TIME PER TICK
2041 004512 012700 000070      MOV      R4, R1          ;GET TICK COUNT
2042 004516 010401
2043 004520 005002          CLR      R2              ;CLEAR SUMMING REGISTERS
2044 004522 005003          CLR      R3
2045 004524 060002      1$:     ADD      R0, R2          ;MULTIPLY TICK COUNT
2046 004526 005503          ADC      R3              ;BY TIME PER TICK

```

```

2047 004530 005301          DEC      R1
2048 004532 001374          BNE     1$
2049
2050 004534 010246          MOV     R2,-(SP)          ;DIVIDE TIME BY 10.
2051 004536 010346          MOV     R3,-(SP)
2052 004540 012746          MOV     #10,-(SP)
2053 004544 004737 000012        JSR     PC,DIVIDE
2054 004550 005726          TST     (SP)+
2055 004552 012637 001012        MOV     @#ATIME          ;DISCARD REMAINDER
2056 004556 113703 001120        MOVB   @#GAP,R3          ;STORE QUOTIENT
2057 004562 006303          ASL     R3                ;GET GAP #
2058 004564 006303          ASL     R3                ;MULTPLY BY 4
2059 004566 023763 001012 001556        CMP     @#ATIME,GTIMBL(R3) ;TO GET AT TABLE ENTRY
2060 004574 101004          BHI     2$                ;CHECK TIME (MAX)
2061 004576 023763 001012 001560        CMP     @#ATIME,GTIMBL+2(R3) ;CHECK TIME (MIN)
2062 004604 101002          BHI     3$
2063 004606 104402          HLT+2  2$                ;REPORT OUT OF RANGE ERROR
2064 004610 000406          BR      100$
2065 004612 032777 000400 174160 3$          BIT     #SW08,@SWR          ;BRANCH IF TIMES NOT WANTED
2066 004620 001402          BEQ     100$
2067 004622 004737 002744        JSR     PC,OUTGAP          ;TYPE GAP TIMES
2068
2069 004626          100$:
2070 004626 004737 002376        JSR     PC,.RESTORE        ;RESTORE REGISTERS FROM THE STACK
2071 004632 000207          RTS     PC                ;RETURN TO TEST
2072
2073          SBTTL      DELAY SUBROUTINES
2074          ;THIS SUBROUTINE CAUSES A DELAY OF 350 MS.
2075 004634 004737 004250        DELAY: JSR     PC,TIMON
2076 004640 010246          MOV     R2,-(SP)          ;SAVE R2 ON THE STACK
2077 004642 012702 004652        MOV     #2$,R2            ;SET RETURN ADDRESS FOR TIMER
2078 004646          1$:
2079 004646 000163 004340        JMP     TIMER(R3)          ;GO TO TIMER & RETURN VIA R2
2080 004652 032704 004000        2$: BIT     #4000,R4
2081 004656 001773          BEQ     1$
2082 004660 012602          MOV     (SP)+,R2          ;RESTORE R2
2083 004662 000207          RTS     PC
2084
2085          ;THIS SUBROUTINE ALLOWS A CALLER SPECIFIED DELAY (UP TO 65MS.)
2086          ;CALL: MOV     DELAY TIME,DELTIM          ;LOAD DELAY TIME (IN US)
2087          ;
2088 004664 005737 001114        DELAYV: JSR     PC,DELAYV
2089 004670 001413          TST     DELTIM            ;BRANCH IF 0 DELAY
2090 004672 004737 004250        BEQ     3$
2091 004676 010246          JSR     PC,TIMON          ;TURN TIMER ON
2092 004700 012702 004710        MOV     R2,-(SP)          ;SAVE R2 ON THE STACK
2093 004704          1$: MOV     #2$,R2            ;SET RETURN ADDRESS FROM TIMER
2094 004704 000163 004340        JMP     TIMER(R3)          ;GO TO TIMER & RETURN VIA R2
2095 004710 023704 001114        2$: CMP     @#DELTIM,R4
2096 004714 101373          BHI     1$
2097 004716 012602          MOV     (SP)+,R2          ;RESTORE R2
2098 004720 000207          3$: RTS     PC
2099
2100          SBTTL      DIVIDE SUBROUTINE
2101          ;THIS SUBROUTINE DIVIDES A DOUBLE PRECISION # AND RETURNS THE RESULT
2102          ;TO THE CALLER ON THE STACK. BOTH DIVIDEND & DIVISOR MUST BE POSITIVE.
    
```

```

2103 ;CALL: MOV LEAST SIGNIFICANT HALF DIVIDEND,-(SP)
2104 ;MOV MOST SIGNIFICANT HALF DIVIDEND,-(SP)
2105 ;MOV #DIVISOR,-(SP)
2106 ;JSR PC,DIVIDE
2107 ;RETURN
2108 ;(SP)=REMAINDER ON STACK
2109 ;2(SP)=QUOTIENT
2110
2111 ;NOTE: THIS SUBROUTINE DESTROYS PREVIOUS CONTENTS OF R0,R1,R2 & R3.
2112
2113 DIVIDE: CLR -(SP) ;SAVE LOC FOR SIGNS
2114 ;MOV #17,-(SP) ;SET ITERATION COUNT
2115 ;MOV 12(SP),R1 ;GET LSH DIVIDEND
2116 ;MOV 10(SP),R0 ;GET MSH DIVIDEND
2117 ;MOV 6(SP),R2 ;GET DIVISOR
2118 ;NEG R2 ;NEGATE DIVISOR
2119 ;CLC ;CLEAR 'C' BIT IN PSW
2120 ;BR 2$
2121 ;1$: ROL R0 ;ROTATE MSH DIVIDEND
2122 ;MOV R0,R3 ;SAVE IN R3
2123 ;ADD R2,R3 ;SUBTRACT DIVISOR FROM MSH DIVIDEND
2124 ;BCC 2$ ;BRANCH IF DIVIDEND > DIVISOR
2125 ;MOV R3,R0 ;SAVE REMAINDER IN R0
2126 ;2$: ROL R1 ;ROTATE LSH DIVIDEND
2127 ;DEC (SP) ;DECREMENT ITERATION COUNT
2128 ;BNE 1$
2129 ;TST (SP)+ ;POP ITERATION COUNTER
2130 ;TST (SP)+ ;POP SIGN CORRECTION
2131 ;MOV R1,6(SP) ;PUSH REMAINDER ON STACK
2132 ;MOV R0,4(SP) ;PUSH QUOTIENT ONTO STACK
2133 ;MOV (SP)+,(SP)
2134 ;RTS PC
2135
2136 ;SBTTL DRIVE SUBROUTINES
2137 ;SUBROUTINE TO CHECK IF DRIVE IS AVAILABLE
2138 ;CALL: MOV# DRIVE#,DRVNUM
2139 ;JSR PC,DRVAVA
2140 ;RETURN: 'C' BIT SET IF NOT AVAILABLE
2141 ;DRVAVA: MOV# @DRVNUM,CS2(R5) ;LOAD DRIVE #
2142 ;BIT #TAP,DT(R5) ;CHECK IF TAPE UNIT
2143 ;BNE 1$
2144 ;JSR PC,RHINIT
2145 ;SEV ;SET 'V' TO IND NOT AVAIL
2146 ;1$: RTS PC ;RETURN
2147
2148 ;SUBROUTINE TO CHECK IF TE16 SLAVE IS AVAILABLE FOR TEST
2149 ;CALL: MOV# DRIVE #,@DRVNUM ;PASS DRIVE # VIA DRVNUM
2150 ;MOV# SLAVE #,@SLVNUM ;PASS SLAVE # VIA SLVNUM
2151 ;JSR PC,SLVAVA ;CALL SUBROUTINE
2152 ;SLVAVA: MOV# @DRVNUM,CS2(R5) ;LOAD DRIVE #
2153 ;MOV# @SLVNUM,TC(R5) ;AND SLAVE #
2154 ;BIT #SPR,DT(R5) ;BRANCH IF SLAVE PRESENT
2155 ;BNE 1$
2156 ;SEV ;SET 'V' TO INDICATE NO SLAVE
2157 ;1$: RTS PC
2158

```



```

2159 ;SUBROUTINE TO INITIALIZE RH CONTROLLER
2160 ;CALL: JSR PC,RHINIT
2161
2162 005070 012765 000040 000010 RHINIT: MOV #40,CS2(R5)
2163 005076 113765 0010C4 000010 MOVB @#DRVNUM,CS2(R5)
2164 005104 005046 CLR -(SP)
2165 005106 113716 001005 MOVB @#SLVNUM,(SP)
2166 005112 012665 000032 MOV (SP)+,TC(R5) ;LOAD SLAVE # INTO TC REG
2167 005116 052765 001700 000032 BIS #NORM11,TC(R5)
2168 005124 000207 RTS PC
2169
2170 ;SUBROUTINE TO WAIT FOR DRIVE READY (DRY)
2171 005126 005027 WAITRDY:CLR (PC)+ ;CLEAR WAIT TIMER
2172 005130 000000 WAITTIM: .WORD 0
2173 005132 105765 000012 IS: TSTB DS(R5) ;WAIT FOR READY TO SET
2174 005136 100406 BMI 2$
2175 005140 005237 005130 INC WAITTIM ;INCREMENT WAIT TIMER
2176 005144 001372 BNE 1$ ;BRANCH IF TIME HAS NOT EXPIRED
2177 005146 000004 014373 TYPE,E.TIMEXP ;TYPE 'TIME EXPIRED WAITING FOR RDY'
2178 005152 000425 BR 99$ ;TAKE ERROR EXIT
2179 005154 032765 002000 000012 2$: BIT #EOT,DS(R5) ;CHECK FOR END OF TAPE
2180 005162 001415 BEQ 3$ ;BRANCH IF NO EOT
2181 005164 000004 013252 TYPE,M.NAM
2182 005170 000004 013756 TYPE,M.EOT ;TYPE 'END OF TAPE'
2183 005174 004737 005232 JSR PC,.REWIND ;REWIND SLAVE
2184 005200 102412 BVS 99$ ;BRANCH IF ERROR ON REWIND
2185 005202 004737 005314 JSR PC,WRITE ;WRITE A RECORD
2186 005206 005215 INC (R5) ;SET 'GO' BIT
2187 005210 004737 005126 JSR PC,WAITRDY ;WAIT FOR READY
2188 005214 000404 BR 99$ ;TAKE ERROR EXIT
2189 005216 032765 040000 000012 3$: BIT #ERR,DS(R5) ;CHECK ERROR EXIT
2190 005224 001401 BEQ 100$
2191 005226 000262 99$: SEV
2192 005230 000207 100$: RTS PC
2193 ;SUBROUTINE TO REWIND A UNIT (DRIVE/SLAVE COMBINATION)
2194 ;CALL MOVB DRIVE #,@#DRVNUM
2195 ; MOVB SLAVE #,@#SLVNUM
2196 ; JSR PC,.REWIND
2197 ;SUBROUTINE RETURNS TO CALLER WITH SELECTED SLAVE AT 'BOT', & 'V' SET IF
2198 ;AN ERROR OCCURS.
2199
2200 005232 004737 005070 .REWIND:JSR PC,RHINIT ;INITIALIZE CONTROLLER
2201 005236 004337 005450 JSR R3,TMCMO ;GO TO TM COMMAND SUBROUTINE
2202 005242 000000 .WORD 0 ;BUS ADDRESS (NOT USED)
2203 005244 000000 .WORD 0 ;WORD COUNT (NOT USED)
2204 005246 000000 .WORD 0 ;FRAME COUNT (NOT USED)
2205 005250 000006 .WORD RWD ;REWIND COMMAND
2206 005252 005215 INC (R5) ;SET 'GO' BIT
2207 005254 032765 000002 000012 1$: BIT #BOT,DS(R5) ;BRANCH IF 'BOT' SET
2208 005262 001005 BNE 2$
2209 005264 032765 040000 000012 BIT #ERR,DS(R5) ;CHECK ERROR BIT
2210 005272 001006 BNE 99$ ;BRANCH IF ERROR BIT SET
2211 005274 000767 BR 1$
2212
2213 005276 032765 020000 000012 2$: BIT #PIP,DS(R5) ;WAIT FOR TAPE MOTION TO STOP
2214 005304 001374 BNE 2$

```

```

2215 005306 000401          BR      100$
2216 005310 000262          99$:  SEV
2217 005312 000207          100$: RTS      PC
2218
2219          ;SUBROUTINE TO WRITE 256. WORD RECORD
2220          ;CALL: JSR      PC,WRITE
2221
2222 005314 004337 005450      WRITE: JSR      R3,TMCMD          ;GO TO TM COMMAND SUBROUTINE
2223 005320 015730              .WORD  WTBUF              ;BUS ADDRESS
2224 005322 177600              .WORD  WRCNT              ;WORD COUNT
2225 005324 177400              .WORD  FRMCNT             ;FRAME COUNT
2226 005326 000060              .WORD  WFWO              ;WRITE FORWARD COMMAND
2227 005330 000207          RTS      PC
2228
2229          ;SUBROUTINE TO READ A 256. WORD RECORD.
2230          ;CALL: JSR      PC,READ
2231
2232 005332 004337 005450      READ:  JSR      R3,R#TMCMD
2233 005336 015730              .WORD  RDBUF              ;ADDRESS OF READ BUFFER
2234 005340 177600              .WORD  WRCNT              ;2'S COMPLEMENT OF WORD COUNT
2235 005342 177400              .WORD  FRMCNT             ;2'S COMPLEMENT OF FRAME COUNT
2236 005344 000070              .WORD  RDFWD              ;READ FORWARD COMMAND
2237 005346 000207          RTS      PC
2238
2239          ;SUBROUTINE TO INITIATE READ REVERSE COMMAND
2240          ;CALL: JSR      PC,REVRD
2241
2242 005350 004337 005450      REVRD: JSR      R3,TMCMD
2243 005354 016330              .WORD  RDBUF+256.        ;ADDRESS OF READ REVERSE BUFFER
2244 005356 177600              .WORD  WRCNT              ;2'S COMPLEMENT OF WORD COUNT
2245 005360 177400              .WORD  FRMCNT             ;2'S COMPLEMENT OF FRAME COUNT
2246 005362 000076              .WORD  RDREV              ;READ REVERSE COMMAND
2247 005364 000207          RTS      PC
2248
2249          ;SUBROUTINE TO SPACE FORWARD 1 RECORD
2250 005366 012765 177777 000006 FWDSPC: MOV      #-1,FC(R5)          ;LOAD RECORD COUNT
2251 005374 012715 000031          MOV      #SPCFWD+1,(R5)      ;LOAD COMMAND
2252 005400 004737 005126          JSR      PC,WAITRDY          ;WAIT FOR READY
2253 005404 000207          RTS      PC              ;RETURN
2254
2255          ;SUBROUTINE TO WRITE A RECORD AND BACK SPACE OVER THE RECORD.
2256 005406 004737 005314      WRT.BK: JSR      PC,WRITE          ;WRITE THE RECORD
2257 005412 005215              INC      (R5)              ;SET 'GO' BIT
2258 005414 004737 005126          JSR      PC,WAITRDY
2259 005420 102412              BVS     2$
2260 005422 012765 177777 000006      MOV      #-1,FC(R5)          ;LOAD RECORD COUNT
2261 005430 012715 000033          MOV      #SPCREV+1,(R5)      ;LOAD COMMAND
2262 005434 004737 005126          JSR      PC,WAITRDY
2263 005440 102402              BVS     2$
2264 005442 004737 004634      1$:  JSR      PC,DELAY          ;WAIT FOR TAPE MOTION TO STOP
2265 005446 000207          2$:  RTS      PC
2266
2267          ;SUBROUTINE TO LOAD A COMMAND
2268          ;CALL: JSR      R3,TMCMD
2269          .WORD  BUS ADDRESS
2270          .WORD  WORD COUNT (2'S COMPLEMENT)
    
```

|      |        |        |        |        |       |                              |                     |
|------|--------|--------|--------|--------|-------|------------------------------|---------------------|
| 2271 |        |        |        | :      | .WORD | FRAME COUNT (2'S COMPLEMENT) |                     |
| 2272 |        |        |        | :      | .WORD | COMMAND                      |                     |
| 2273 |        |        |        |        |       |                              |                     |
| 2274 | 005450 | 012365 | 000004 | TMCMD: | MOV   | (R3)+,BA(R5)                 | ;LOAD BUS ADDRESS   |
| 2275 | 005454 | 012365 | 000002 |        | MOV   | (R3)+,WC(R5)                 | ;LOAD WORD COUNT    |
| 2276 | 005460 | 012365 | 000006 |        | MOV   | (R3)+,FC(R5)                 | ;LOAD FRAME COUNT   |
| 2277 | 005464 | 012315 |        |        | MOV   | (R3)+,(R5)                   | ;LOAD COMMAND       |
| 2278 | 005466 | 000203 |        |        | RTS   | R3                           | ;RETURN             |
| 2279 |        |        |        |        |       |                              |                     |
| 2280 |        |        |        |        |       |                              |                     |
| 2281 |        |        |        |        |       |                              |                     |
| 2282 |        |        |        |        |       |                              |                     |
| 2283 | 005470 | 016503 | 000030 | SNPT:  | MOV   | SN(R5),R3                    |                     |
| 2284 | 015474 | 012701 | 001144 |        | MOV   | #00IGITS,R1                  |                     |
| 2285 | 015478 | 000303 |        |        | SWAB  | R3                           |                     |
| 2286 | 015482 | 016503 |        |        | ROR   | R3                           |                     |
| 2287 | 015486 | 016503 |        |        | ROR   | R3                           |                     |
| 2288 | 015490 | 016503 |        |        | ROR   | R3                           |                     |
| 2289 | 015494 | 016503 |        |        | ROR   | R3                           |                     |
| 2290 | 015498 | 042703 | 177760 |        | BIC   | #177760,R3                   | ;GET FIRST DIGIT    |
| 2291 | 015502 | 052703 | 000260 |        | BIS   | #260,R3                      |                     |
| 2292 | 005522 | 110321 |        |        | MOVB  | R3,(R1)+                     | ;FILL FIRST DIGIT   |
| 2293 | 005524 | 016503 | 000030 |        | MOV   | SN(R5),R3                    |                     |
| 2294 | 005528 | 000303 |        |        | SWAB  | R3                           |                     |
| 2295 | 005532 | 042703 | 177760 |        | BIC   | #177760,R3                   |                     |
| 2296 | 005536 | 052703 | 000260 |        | BIS   | #260,R3                      |                     |
| 2297 | 005540 | 110321 |        |        | MOVB  | R3,(R1)+                     | ;GET SECOND DIGIT   |
| 2298 | 005544 | 016503 | 000030 |        | MOV   | SN(R5),R3                    |                     |
| 2299 | 005548 | 006003 |        |        | ROR   | R3                           |                     |
| 2300 | 005552 | 006003 |        |        | ROR   | R3                           |                     |
| 2301 | 005556 | 006003 |        |        | ROR   | R3                           |                     |
| 2302 | 005560 | 006003 |        |        | ROR   | R3                           |                     |
| 2303 | 005564 | 042703 | 177760 |        | BIC   | #177760,R3                   |                     |
| 2304 | 005568 | 052703 | 000260 |        | BIS   | #260,R3                      |                     |
| 2305 | 005572 | 110321 |        |        | MOVB  | R3,(R1)+                     | ;GET THIRD DIGIT    |
| 2306 | 005576 | 016503 | 000030 |        | MOV   | SN(R5),R3                    |                     |
| 2307 | 005580 | 042703 | 177760 |        | BIC   | #177760,R3                   |                     |
| 2308 | 005584 | 052703 | 000260 |        | BIS   | #260,R3                      |                     |
| 2309 | 005588 | 110321 |        |        | MOVB  | R3,(R1)+                     | ;GET FOURTH DIGIT   |
| 2310 | 005592 | 105011 |        |        | CLRB  | (R1)                         |                     |
| 2311 | 005596 | 000004 | 001144 |        | TYPE, | 00IGITS                      | ;TYPE SERIAL NUMBER |
| 2312 | 005600 | 000207 |        |        | RTS   | PC                           | ;RETURN             |
| 2313 |        |        |        |        |       |                              |                     |

```

2314 .SBTTL PROGRAM INITIALIZATION
2315 005620 012706 000600 INIT: MOV #STKPTR, SP ;SET STACK PTR
2316 005624 005037 001264 CLR @#INBUF
2317
2318 005630 013746 000006 MOV @#6, -(SP) ;SAVE VECTORS
2319 005634 013746 000004 MOV @#4, -(SP)
2320 005640 012737 005660 000004 MOV @#61$, @#4 ;SET UP FOR TIMEOUT
2321 005646 022777 177777 173124 CMP @-1, @SWR ;REFERENCE HARDWARE SWITCH REGISTER
2322 005654 001402 BEQ 60$
2323 005656 000404 BR 62$
2324 005660 022626 61$: CMP (SP)+, (SP)+ ;ADJUST STACK
2325 005662 012737 000176 001000 60$: MOV @SWREG, SWR ;POINT TO SOFTWARE SWITCH REG
2326 005670 012637 000004 62$: MOV (SP)+, @#4 ;RESTORE VECTORS
2327 005674 012637 000006 MOV (SP)+, @#6
2328 005700 105037 001124 CLRB @#PRGFLG ;CLEAR PROGRAM FLAG
2329 005704 105037 001130 CLRB @#ASFLG ;CLEAR ASK FLAG
2330 005710 105037 001127 CLRB @#PSCNT ;SET PASS COUNT = 0
2331 005714 005027 CLR (PC)+ ;CLEAR CHAIN INDICATOR
2332 005716 000000 CHNFLG: .WORD 0 ;CHAIN MODE INDICATOR
2333 ;1/0 = CHAIN/NOT CHAIN MODE
2334 005720 022737 012642 000042 CMP @SENDAD, @#42 ;BRANCH IF LOADED VIA ACT11 CHAIN MODE
2335 005726 001404 BEQ 50$
2336 005730 005737 000042 TST @#42 ;BRANCH IF IN DUMP MODE
2337 005734 001413 BEQ 52$
2338 005736 000406 BR 51$
2339 005740 012737 000176 001000 50$: MOV @SWREG, SWR ;INVOKE SOFTWARE SWR
2340 005746 012777 100000 173024 MOV @100000, @SWR ;WITH HALT ON ERROR SET
2341 005754 005237 005716 51$: INC CHNFLG ;SET CHNFLG = CHAIN MODE
2342 005760 000137 006056 JMP 5$ ;GO TO CHAIN ADDRESS
2343 005764 52$:
2344 005764 122737 000006 000041 CMPB @#6, @#41 ;BRANCH IF NOT LOADED VIA TMDP
2345 005772 001002 BNE 1$
2346 005774 000004 013455 TYPE, I.REM ;ADVISE USER TO REMOVE TMDP
2347 006000 000004 013252 1$: TYPE, M.NAM ;TYPE TITLE
2348 006004 105037 013252 CLRB M.NAM ;DO NOT TYPE TITLE ON RESTART
2349 006010 000004 013522 TYPE, I.REG ;ASK USER TO TYPE CONT BASE ADRS
2350 006014 013702 001010 MOV @#TMBASE, R2 ;GET CURRENT CONT BASE ADDRESS
2351 006020 004737 002426 JSR PC, TYPOCT ;AND TYPE IT
2352 006024 000004 001410 TYPE, SPACE
2353 006030 004737 003272 JSR PC, INPUT ;GET USER INPUT
2354 006034 122737 000015 001264 CMPB @CR, @#INBUF ;DO NOT CHANGE CURRENT VALUE
2355 006042 001405 BEQ 5$ ;IF USER TYPES <CR>
2356 006044 004737 003056 4$: JSR PC, CNVTA0 ;CONVERT ASCII TO OCTAL
2357 006050 013737 001116 001010 MOV @#OCTALO, @#TMBASE ;SET NEW ADDRESS
2358 006056 013705 5$: MOV @#TMBASE, R5
2359
2360 ;ROUTINE TO CHECK IF CONTROLLER (RH11) IS AVAILAABLE
2361 006062 000261 SEC ;SET 'C' IN PSW
2362 006064 005715 TST (R5) ;BRANCH IF CONTROLLER AVAIL
2363 006066 103003 BCC 6$
2364 006070 000004 014015 TYPE, E.NCON
2365 006074 000651 BR INIT
2366 006076 012737 003542 000004 6$: MOV @ERRTRP, @#ERRVEC ;SET ERROR TRAP VECTOR

```

```

2367
2368 006104 105037 001123
2369 006110 012701 001154
2370 006114 012700 000004
2371 006120 005021
2372 006122 005300
2373 006124 001375
2374 006126 005737 005716
2375 006132 001014
2376 006134 000004 013567
2377 006140 004737 003272
2378 006144 012700 001264
2379 006150 122710 000101
2380 006154 001403
2381 006156 122710 000015
2382 006162 001013
2383 006164 110637 001124
2384 006170 012701 001154
2385 006174 012700 000004
2386 006200 012721 177777
2387 006204 005300
2388 006206 001374
2389 006210 000417
2390
2391
2392 006212 122710 000015
2393 006216 001414
2394 006220 121027 000054
2395 006224 001001
2396 006226 105720
2397 006230 112001
2398 006232 042701 177770
2399 006236 112761 177777 001154
2400 006244 000240
2401 006246 000761
2402
2403
2404 006250 005000
2405 006252 105760 001154
2406 006256 001005
2407 006260 005200
2408 006262 122700 000010
2409 006266 001371
2410 006270 000424
2411 006272 110037 001004
2412 006276 004737 005012
2413 006302 102366
2414 006304 105737 001124
2415 006310 001011
2416 006312 000004 014062
2417 006316 116037 001132 014114
2418 006324 000004 014114
2419 006330 110637 001123
2420 006334 105060 001154
2421 006340 000747
2422 006342 105737 001123

; ROUTINE TO GET TMO3 DRIVES USER DESIRES TO TEST
DRIVES: CLRB 2#ERFLG ; CLEAR ERROR FLAG
MOV 2#DRVTBL,R1 ; MARK ALL DRIVES AS NOT TO
MOV 2#4,RO ; BE TESTED. A '0' INDICATES
1$: CLR (R1)+ ; THAT A DRIVE IS NOT TO BE
DEC RO ; TESTED
BNE 1$
TST 2#CNFLG ; BRANCH IF IN CHAIN MODE
BNE 2$
TYPE,I,DRVS
JSR PC,INPUT ; GET USER INPUT
MOV 2#INBUF,RO
CMPB 2#'A,(RO) ; IF USER RESPONDS WITH 'A' OR
BEQ 2$ ; (CR) THEN ALL AVAILABLE DRIVES
CMPB 2#CR,(RO) ; ARE TO BE TESTED
BNE 4$
2$: MOVB SP,PRGFLG ; SET FLAG TO IND ALL DRIVES
MOV 2#DRVTBL,R1 ; MARK ALL DRIVES TO BE TESTED
MOV 2#4,RO ; A '-1' INDICATES THAT A DRIVE
3$: MOV 2#-1,(R1)+ ; IS TO BE TESTED
DEC RO
BNE 3$
BR 2#CHKDRV ; GO CHECK DRIVE AVAILABILITY

; GET USER SELECTED DRIVES AND MARK EACH DRIVE SELECTED TO BE TESTED
4$: CMPB 2#CR,(RO)
BEQ 2#CHKDRV
CMPB (RO),2#',
BNE 5$ ; CHECK IF 'COMMA'
TSTB (RO)+ ; STEP PTR PAST 'COMMA'
5$: MOVB (RO)+,R1
BIC 2#177770,R1
MOVB 2#-1,DRVTBL(R1)
NOP
BR 4$

; ASCERTAIN THAT DRIVES (TMO3'S) SPECIFIED ARE AVAILABLE
CHKDRV: CLR RO ; A (0) IN DRVTBL(RO) INDICATES
1$: TSTB DRVTBL(RO) ; THE DRIVE IS NOT TO BE TESTED
BNE 3$ ; A '1' INDICATES TO BE TESTED
2$: INC RO
CMPB 2#8.,RO
BNE 1$
BR 5$
3$: MOVB RO,2#DRVNUM ; GET DRIVE #
JSR PC,2#DRVAVA ; AND CHECK IF AVAILABLE
BVC 2$ ; 'V' BIT SET INDICATES NOT AVAIL
TSTB 2#PRGFLG ; DO NOT TYPE NOT AVAILABLE
BNE 4$ ; MESSAGE IF ALL SELECTED
TYPE,E,NORV
MOVB DIGTAB(RO),2#E.NAVA ; SET DRIVE # IN MESSAGE
TYPE,E,NAVA
4$: MOVB SP,2#ERFLG ; SET 'ERROR' FLAG
CLRB DRVTBL(RO) ; MARK DRIVE UNAVAILABLE
BR 2$ ; CHECK NEXT DRIVE
5$: TSTB 2#ERFLG ; GO GET SLAVES IF NO ERROR

```

```

223 006346 001256          BNE      DRIVES          ;ELSE ASK USER TO RETYPE DRIVES
;ROUTINE TO GET SLAVES (TE16'S) USER DESIRES TO TEST
SLAVES: CLR      B          ;CLEAR ERROR INDICATOR
        MOV      #ERFLG          ;MARK ALL SLAVES (64.) AS NOT
        MOV      #SLVTBL,R1      ;TO BE TESTED.A 0 INDICATES THAT
        MOV      #32,R0         ;A DRIVE'S SLAVE IS NOT TO BE
1$:     CLR      (R1)+          ;TESTED
        DEC      R0
        BNE     1$
        MOV      #SLVTBL,R1      ;R1 POINTS TO DRIVE'S SLAVE
2$:     TSTB    DRVTBL(R0)      ;BRANCH IF DRIVE IS TO BE TESTED
        BNE     4$              ;& IS AVAILABLE
        ADD     #8.,R1          ;STEP SLAVE PTR TO NEXT DRIVE'S
3$:     INC     R0              ;SLAVES AND INCREMENT DRIVE #
        CMPB   #8.,R0          ;CHECK ALL DRIVES
        BNE     2$            ;AND WHEN ALL DRIVES CHECKED
        BR     CHKSLV          ;GO CHECK SLAVE AVAILABILITY

        TSTB   #PRGFLG         ;BRANCH IF USER SELECTED ALL
4$:     BNE     5$              ;DRIVES
        MOVB   R0,DRVNUM        ;GET DRIVE #
        MOVB   DIGTAB(R0),#I.DRV ;PREPARE USER ACTION MESSAGE
        TYPE, I.SLVS
        JSR    PC,INPUT        ;GET USER INPUT
        MOV    #INBUF,R3       ;SET PTR TO USER INPUT
        CMPB  #'A,(R3)         ;AN 'A' OR <CR> AS FIRST CHAR
        BEQ   5$              ;INDICATES TEST ALL SLAVES
        CMPB  #CR,(R3)
        BNE   7$
5$:     MOVB   SP,#PRGFLG      ;SET 'ALL' INDICATOR
        MOV    #SLVTBL,R1      ;MARK ALL SLAVES FOR ALL
        MOV    #32,R0          ;DRIVES AS TO BE TESTED
6$:     MOV    #-1,(R1)+
        DEC   R0
        BNE   6$
        TSTB  #PRGFLG         ;BRANCH IF ALL WAS SELECTED
7$:     BNE   CHKSLV
        CMPB  #CR,(R3)         ;GET USER SELECTED SLAVES FOR
        BEQ   3$              ;DRIVE
        CMPB  (R3),#',
        BNE   8$
        TSTB  (R3)+
8$:     MOVB  (R3)+,R4          ;AND MARK SELECED SLAVE
        BIC   #177770,R4      ;AS TO BE TESTED
        ADD   R1,R4
        MOVB  #-1,(R4)
        BR    7$

;ASCERTAIN THAT SLAVES (TE16'S) SELECTED ARE AVAILABLE
CHKSLV: CLR      R0          ;R0 WILL CONTAIN THE DRIVE #
        CLR    R1              ;AND R1 THE SLAVE #
        MOV    #SLVTBL,R2      ;SET PTR TO SLAVE TABLE
1$:     TSTB   DRVTBL(R0)      ;BRANCH IF DRIVE SELECTED
        BNE   3$              ;& AVAILABLE FOR TEST
2$:     INC    R0              ;INCREMENT DRIVE #

```

013656

```

2579 006600 062702 000010      ADD    #8.,R2      ;STEP SLAVE PTR TO NEXT DRIVE'S
2580 006604 022700 000010      CMP    #8.,R0      ;SLAVES. BRANCH TO 1$ IF NOT ALL
2581 006610 001367              BNE    1$          ;DRIVES CHECKED OTHERWISE EXIT
2582 006612 000437              BR     8$
2583
2584 006614 005001      3$:   CLR    R1      ;SET SLAVE # 0
2585 006616 105712      4$:   TSTB   (R2)    ;BRANCH IF DRIVE'S SLAVE IS SEL-
2586 006620 001006              BNE    6$          ;ECTED FOR TEST
2587 006622 005201      5$:   INC    R1      ;INCREMENT SLAVE #
2588 006624 005202              INC    R2          ;STEP PTR TO NEXT SLAVE
2589 006626 022701 000010      CMP    #8.,R1    ;GO TO 4$ IF ALL SLAVES NOT
2590 006632 001371              BNE    4$          ;CHECKED
2591 006634 000760              BR     2$          ;OTHERWISE GO TO 2$ ABOVE
2592
2593 006636 110037 001004      6$:   MOVB   R0,@#DRVNUM ;PASS DRIVE & SLAVE #
2594 006642 110137 001005      MOVB   R1,@#SLVNUM
2595 006646 004737 005040      JSR    PC,@#SLVAVA ;AND CHECK IF AVAILABLE
2596 006652 102363              BVC    5$          ;'V' SET INDICATES ERROR
2597 006654 105737 001124      TSTB   @#PRGFLG  ;DO NOT TYPE ERROR MSG IF ALL
2598 006660 001012              BNE    7$          ;SLAVES SELECTED
2599 006662 116037 001132 014104      MOVB   DIGTAB(R0),@#E.DRV ;ICATES ERROR. PREPARE ERROR
2600 006670 116137 001132 014114      MOVB   DIGTAB(R1),@#E.NAVA ;MESSAGE
2601 006676 000004 014076      TYPE,E.NSLV
2602 006702 110637 001123      MOVB   SP,@#ERFLG ;SET ERROR INDICATOR
2603 006706 105012      7$:   CLRB   (R2)    ;CLEAR SLAVE TABLE ENTRY
2604 006710 000744              BR     5$          ;GET NEXT SLAVE
2605
2606 006712 105737 001123      8$:   TSTB   @#ERFLG  ;BRANCH IF ERROR
2607 006716 001214              BNE    SLAVES     ;ASK USER TO RETYPE SLAVES
2608 006720 012737 003542 000004      10$:  MOV    #ERRTRP,@#ERRVEC
2609
2610      ;SCAN DRIVE AND SLAVE TABLE FOR DRIVE/SLAVE COMBINATION TO TEST.
2611      ;RESTART ADDRESS--PROGRAM STARTS HERE WHEN START ADDRESS = 210 AND
2612      ;AFTER EACH PASS
2613 006726 105037 001004      RSTRT: CLRB   @#DRVNUM ;SET DRIVE AND SLAVE # 0
2614 006732 105037 001005      CLRB   @#SLVNUM
2615 006736 012737 001164 001006      MOV    #SLVTBL,@#SLVPTR ;SET PTR TO SLAVE TABLE
2616 006744 105037 001125      CLRB   @#UNTFND   ;CLEAR 'UNIT FOUND' IND.
2617
2618      ;PROGRAM RESTARTS HERE AFTER EACH DRIVE/SLAVE HAS BEEN TESTED.
2619 006750 113700 001004      BEGIN: MOVB   @#DRVNUM,R0 ;GET DRIVE #
2620 006754 113701 001005      MOVB   @#SLVNUM,R1  ;AND SLAVE #
2621 006760 013702 001006      MOV    @#SLVPTR,R2  ;GET SLAVE PTR
2622 006764 122737 000006 000041      CMPB   #6,@#41     ;BRANCH IF LOADED VIA TMDP
2623 006772 001001              BNE    1$
2624 006774 105012              CLRB   (R2)        ;SET DRIVE #0, SLAVE #0 NOT TO
2625                          ;BE TESTED.
2626 006776 105760 001154      1$:   TSTB   DRVTBL(R0) ;BRANCH IF DRIVE AVAIL TO TEST
2627 007002 001011              BNE    3$
2628 007004 005001              CLR    R1          ;CLEAR SLAVE #
2629 007006 062702 000010      ADD    #8.,R2      ;AND STEP PTR TO NEXT DRIVE'S
2630 007012 005200      2$:   INC    R0          ;SLAVES AND INCREMENT DRIVE #
2631 007014 022700 000010      CMP    #8.,R0      ;EXIT TEST IF ALL DRIVES
2632 007020 001366              BNE    1$          ;CHECKED OTHERWISE CONTINUE
2633 007022 000137 012570      JMP    @#END       ;SCAN FOR NEXT 'UNIT'
2634

```

|      |        |        |        |
|------|--------|--------|--------|
| 2535 | 007026 | 105712 |        |
| 2536 | 007030 | 001007 |        |
| 2537 | 007032 | 005202 |        |
| 2538 | 007034 | 005201 |        |
| 2539 | 007036 | 122701 | 000010 |
| 2540 | 007042 | 001371 |        |
| 2541 | 007044 | 005001 |        |
| 2542 | 007046 | 000761 |        |
| 2543 |        |        |        |
| 2544 | 007050 | 110637 | 001125 |
| 2545 | 007054 | 110037 | 001004 |

```

3$:  TSTB   (R2)
      BNE   4$
      INC   R2
      INC   R1
      CMPB  #8.,R1
      BNE   3$
      CLR   R1
      BR    2$
  
```

```

; BRANCH IF SLAVE ON DRIVE IS
; AVAILABLE THEREWISE STEP
; PTR TO NEXT SLAVE
; INCREMENT SLAVE #
; UNTIL ALL SLAVES CHECKED
; WHEN ALL SLAVES CHECKED
; SET SLAVE # 0
; AND CONTINUE SCAN
  
```

```

4$:  MOVB   SP,2#UNTFND
      MOVB  RO,2#DRVNUM
  
```

```

; INDICATE THAT A 'UNIT' IS FOUND
; SET DRIVE 3
  
```



DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER  
 DZTEEA.P11 31-MAR-77 00:00

MACY11 30(1046) 27-JUN-77 17:03 PAGE 59  
 PROGRAM INITIALIZATION

|      |        |        |        |        |     |             |                           |
|------|--------|--------|--------|--------|-----|-------------|---------------------------|
| 2546 | 007060 | 110137 | 001005 |        | MOV | R1, @SLVNUM | ; SET SLAVE #             |
| 2547 | 007064 | 010237 | 001006 |        | MOV | R2, @SLVPTR | ; SAVE SLAVE PTR          |
| 2548 |        |        |        |        |     |             |                           |
| 2549 | 007070 | 105737 | 001130 |        | 5S: | TST         | @ASFLG                    |
| 2550 | 007074 | 001034 |        |        |     | BNE         | 7S                        |
| 2551 | 007076 | 112737 | 000001 | 001130 |     | MOV         | #1, ASFLG                 |
| 2552 | 007104 | 005737 | 005716 |        |     | TST         | CHNFLG                    |
| 2553 | 007110 | 001026 |        |        |     | BNE         | 7S                        |
| 2554 | 007112 | 105037 | 001124 |        |     | CLRB        | @PRGFLG                   |
| 2555 | 007116 | 000004 | 013723 |        |     | TYPE, I.SPD | ; CLEAR PROGRAM INDICATOR |
| 2556 | 007122 | 004737 | 003272 |        |     | JSR         | PC, INPUT                 |
| 2557 | 007126 | 012703 | 001264 |        |     | MOV         | #INBUF, R3                |
| 2558 | 007132 | 122713 | 000015 |        |     | CMPB        | #CR, (R3)                 |
| 2559 | 007136 | 001405 |        |        |     | BEQ         | 6S                        |
| 2560 | 007140 | 132713 | 000001 |        |     | BITB        | #1, (R3)                  |
| 2561 | 007144 | 001402 |        |        |     | BEQ         | 6S                        |
| 2562 | 007146 | 111337 | 001124 |        |     | MOV         | (R3), @PRGFLG             |
| 2563 | 007152 | 022737 | 000176 | 001000 | 6S: | CMP         | #SWREG, SWR               |
| 2564 | 007160 | 001002 |        |        |     | BNE         | 7S                        |
| 2565 | 007162 | 004737 | 002020 |        |     | JSR         | PC, GTSWR                 |
| 2566 | 007166 |        |        |        | 7S: |             |                           |
| 2567 |        |        |        |        |     |             |                           |
| 2568 |        |        |        |        |     |             |                           |
| 2569 |        |        |        |        |     |             |                           |
| 2570 | 007166 | 013705 | 001010 |        |     | MOV         | @TMBASE, R5               |
| 2571 | 007172 | 010500 |        |        |     | MOV         | R5, R0                    |
| 2572 | 007174 | 062700 | 000006 |        |     | ADD         | #FC, R0                   |
| 2573 | 007200 | 010501 |        |        |     | MOV         | R5, R1                    |
| 2574 | 007202 | 062701 | 000012 |        |     | ADD         | #DS, R1                   |
| 2575 | 007206 | 012703 | 004340 |        |     | MOV         | #TIMER, R3                |
| 2576 | 007212 | 105037 | 001121 |        |     | CLRB        | @ITCNT                    |
| 2577 | 007216 | 052737 | 000100 | 177560 |     | BIS         | #100, @TKS                |
| 2578 |        |        |        |        |     |             |                           |
| 2579 |        |        |        |        |     |             |                           |
| 2580 |        |        |        |        |     |             |                           |
| 2581 |        |        |        |        |     |             |                           |
| 2582 |        |        |        |        |     |             |                           |
| 2583 | 007224 | 004737 | 005232 |        |     | JSR         | PC, REWIND                |
| 2584 | 007230 | 102474 |        |        |     | BVS         | 99S                       |
| 2585 | 007232 | 004737 | 005314 |        |     | JSR         | PC, WRITE                 |
| 2586 | 007236 | 005215 |        |        |     | INC         | (R5)                      |
| 2587 | 007240 | 004737 | 005126 |        |     | JSR         | PC, WAITRDY               |
| 2588 | 007244 | 102466 |        |        |     | BVS         | 99S                       |
| 2589 | 007246 | 005737 | 005716 |        |     | TST         | CHNFLG                    |
| 2590 | 007252 | 001064 |        |        |     | BNE         | 100S                      |
| 2591 | 007254 | 117702 | 171520 |        |     | MOV         | @SWR, R2                  |
| 2592 | 007260 | 042702 | 177740 |        |     | BIC         | #177740, R2               |
| 2593 | 007264 | 001421 |        |        |     | BEQ         | 2S                        |
| 2594 | 007266 | 000004 | 014235 |        |     | TYPE, E.HOR | ; GET SWITCHES            |
| 2595 | 007272 | 004737 | 002426 |        |     | JSR         | PC, TYPOCT                |
| 2596 | 007276 | 006302 |        |        |     | ASL         | R2                        |
| 2597 | 007300 | 016237 | 001656 | 007310 |     | MOV         | NAMPTR(R2), 1S            |
| 2598 | 007306 | 000004 |        |        |     | TYPE        |                           |
| 2599 | 007310 | 000000 |        |        | 1S: | .WORD       | 0                         |
| 2600 | 007312 | 000004 | 001374 |        |     | TYPE, CRLF  |                           |
| 2601 | 007316 | 016237 | 001736 | 001002 |     | MOV         | TSTBL(R2), @SCPADR        |

; SET SCOPE ADDRESS FOR TEST



2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674

```

;SBTTL START OF TESTS
;TEST 001 - WRITE FROM BOT
;THIS TEST WILL MEASURE ACCELERATION DELAY REQUIRED TO
;MOVE THE TAPE APPROXIMATELY SEVEN (7) INCHES FORWARD
;FROM DEAD STOP BEFORE STARTING TO TRANSFER DATA.

;THIS TEST MEASURES TIME FROM 'GO'=1 TO 'ACCL'=0.
†ST001: MOV# 01,2†STNUM ;SET TEST #
MOV #1$,R2 ;SET RETURN PC FROM TIMER
JSR PC,REWIND ;REWIND SLAVE
BVS 99$ ;BRANCH IF ERROR ON REWIND
JSR PC,WRITE ;GO SETUP WRITE COMMAND
JSR PC,TIMON ;TURN TIMER ON
INC (R5) ;SET 'GO' BIT

1$: TST TC(R5) ;BRANCH WHEN 'ACCL'=0
BPL 2$
JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2

2$: JSR PC,WAITRDY ;WAIT FOR COMMAND TO FINISH
BVS 99$ ;BRANCH IF ERROR
JSR PC,TIMOK ;GO CHECK TIME
BR 100$

99$: HLT
100$: SCOPE

;TEST 002 - WRITE START
;THIS TST MEASURES TIME FROM 'GO'=1 TO 'ACCL'=0.
†ST002: MOV# 02,2†STNUM ;SET TEST # 2
JSR PC,WRITE ;INITIATE WRITE COMMAND
MOV #1$,R2 ;SET RETURN PC FROM TIMER
JSR PC,TIMON
INC (R5) ;SET 'GO' BIT

1$: TST TC(R5) ;BRANCH WHEN 'ACCL'=0
BPL 2$
JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2

2$: JSR PC,WAITRDY ;WAIT FOR READY
BVS 99$ ;BRANCH IF ERROR
JSR PC,TIMOK ;GO CHECK TIME RECORDED
BR 100$ ;EXIT VIA SCOPE

99$: HLT ;REPORT ERROR
100$: SCOPE

;TEST 003- WRITE SHUTDOWN
;THIS TEST MEASURES TIME FROM 'FC REG'=0 TO 'SWDN'=1.
†ST003: MOV# 03,2†STNUM ;SET TEST#3
JSR PC,WRITE ;INITIATE WRITE COMMAND
INC (R5) ;SET 'GO' BIT

1$: TST (R0) ;BRANCH WHEN WRITING FINISHED
BEQ 2$
BIT #ERR,(R1) ;MONITOR ERROR BIT
BNE 99$

```

040000

```

2675 007622 000772          BR      1$
2676
2677 007624          2$:
2678 007624 004737 004250      JSR      PC,TIMON          ;TURN TIMER ON
2679 007630 010702          MOV      PC,R2            ;LOAD RETURN PC FROM TIMER
2680 007632 032711 000020      3$:      BIT      #SDWN,(R1)    ;BRANCH WHEN DS <SDWN> SETS
2681 007636 001002          BNE     4$
2682 007640 000163 004340      JMP      TIMER(R3)        ;GO TO TIMER & RETURN VIA R2
2683
2684 007644 004737 005126      4$:      JSR      PC,WAITRDY       ;WAIT FOR READY
2685 007650 102403          BVS     99$
2686 007652 004737 004376      JSR      PC,TIMOK         ;GO CHECK TIME RECORDED
2687 007656 000401          BR      100$
2688 007660 104400          99$:     HLT
2689 007662 104000          100$:    SCOPE           ;REPORT ERROR
2690
2691          ;TEST 004 - WRITE SETTLEDOWN
2692          ;THIS TEST MEASURES TIME FROM 'SDWN'=1 TO 'SDWN'=0.
2693 007664 112737 000004 001122 1$T004: MOVB     #4,#1$TNUM
2694 007672 004737 005314          JSR      PC,WRITE
2695 007676 005215          INC      (R5)            ;SET 'GO' BIT
2696
2697 007700 005710          1$:      TST      (R0)          ;BRANCH WHEN WRITING FINISHED
2698 007702 001404          BEQ     2$
2699 007704 032711 040000      BIT      #ERR,(R1)       ;CHECK ERROR BIT
2700 007710 001026          BNE     99$
2701 007712 000772          BR      1$
2702
2703 007714 032711 000020      2$:      BIT      #SDWN,(R1)    ;WAIT FOR ASSERTION OF 'SDWN'
2704 007720 001004          BNE     3$
2705 007722 032711 040000      BIT      #ERR,(R1)       ;MONITOR ERROR BIT
2706 007726 001017          BNE     99$
2707 007730 000771          BR      2$
2708
2709 007732          3$:
2710 007732 004737 004250      JSR      PC,TIMON          ;TURN TIMER ON
2711 007736 010702          MOV      PC,R2            ;SET RETURN PC FROM TIMER
2712 007740 032711 000020      BIT      #SDWN,(R1)    ;BRANCH WHEN SDWN CLEARS
2713 007744 001402          BEQ     5$
2714 007746 000163 004340      JMP      TIMER(R3)        ;GO TO TIMER & RETURN VIA R2
2715
2716 007752 004737 005126      5$:      JSR      PC,WAITRDY       ;WAIT FOR READY
2717 007756 102403          BVS     99$
2718 007760 004737 004376      JSR      PC,TIMOK
2719 007764 000401          BR      100$
2720
2721 007766 104400          99$:     HLT
2722 007770 104000          100$:    SCOPE
2723
2724          ;TEST 005 - READ FROM BOT
2725          ;THIS TEST MEASURES TIME FROM 'GO'=1 TO 'ACCL'=0.
2726 007772 112737 000005 001122 1$T005: MOVB     #5,#1$TNUM
2727 010000 004737 005232          JSR      PC,.REWIND       ;SET TEST #5
2728 010004 102422          BVS     99$              ;REWIND SLAVE
2729 010006 004737 005332          JSR      PC,READ          ;BRANCH IF ERROR ON REWIND
2730 010012 012702 010024          MOV      #1$,R2          ;SET RETURN PC FROM TIMER
    
```

```

2731 010016 004737 004250          JSR   PC,TIMON          ;TURN TIMER ON
2732 010022 005215                   INC   (R5)              ;SET 'GO' BIT
2733
2734 010024 005765 000032          1$:   TST   TC(R5)          ;BRANCH WHEN 'ACCL' RESETS
2735 010030 100002                   BPL   2$
2736 010032 000163 004340          JMP   TIMER(R3)        ;GO TO TIMER & RETURN VIA R2
2737
2738 010036 004737 005126          2$:   JSR   PC,WAITRDY      ;WAIT FOR READY
2739 010042 102403                   BVS   99$              ;BRANCH IF ERROR
2740 010044 004737 004376          JSR   PC,TIMOK         ;CHECK RECORDED TIME
2741 010050 000401                   BR    100$
2742
2743 010052 104400          99$:   HLT
2744 010054 104000          100$:  SCOPE
2745
2746          ;TEST 006 - READ START
2747          ;THIS TEST MEASURES TIME FROM 'GO'=1 TO 'ACCL'=0.
2748 010056 112737 000006 001122  TST006: MOVB   #6,#1STNUM ;SET TEST #6
2749 010064 004737 005406          JSR   PC,WRT.BK        ;WRITE A RECORD & BACK SPACE
2750 010070 102422                   BVS   99$
2751 010072 004737 005332          JSR   PC,READ
2752 010076 012702 010110          MOV   #1$,R2          ;SET RETURN PC FROM TIMER
2753 010102 004737 004250          JSR   PC,TIMON        ;TURN TIMER ON
2754 010106 005215                   INC   (R5)              ;SET 'GO' BIT
2755
2756 010110 005765 000032          1$:   TST   TC(R5)          ;BRANCH WHEN 'ACCL' RESETS
2757 010114 100002                   BPL   2$
2758 010116 000163 004340          JMP   TIMER(R3)        ;GO TO TIMER & RETURN VIA R2
2759
2760 010122 004737 005126          2$:   JSR   PC,WAITRDY      ;WAIT FOR READY
2761 010126 102403                   BVS   99$              ;BRANCH IF ERROR
2762 010130 004737 004376          JSR   PC,TIMOK         ;CHECK RECORDED TIME
2763 010134 000401                   BR    100$
2764
2765 010136 104400          99$:   HLT
2766 010140 104000          100$:  SCOPE
2767
2768          ;TEST 007 - READ SHUTDOWN
2769          ;THIS TEST MEASURES TIME FROM 'FC REG'=FRAME COUNT TO 'SDWN'=1.
2770 010142 112737 000007 001122  TST007: MOVB   #7,#1STNUM ;SET TEST #7
2771 010150 004737 005406          JSR   PC,WRT.BK        ;WRITE A RECORD & BACK SPACE
2772 010154 102430                   BVS   99$              ;BRANCH IF ERROR
2773 010156 004737 005332          JSR   PC,READ
2774 010162 005215                   INC   (R5)              ;SET 'GO' BIT
2775
2776 010164 022710 000400          1$:   CMP   #-FRMCNT,(R0)    ;WAIT FOR FRAME COUNT TO
2777 010170 001404                   BEQ   2$               ;= # OF FRAMES WRITTEN
2778 010172 032711 040000          BIT   #ERR,(R1)        ;MONITOR ERROR BIT
2779 010176 001017                   BNE   99$
2780 010200 000771                   BR    1$
2781
2782          2$:
2783 010202 004737 004250          JSR   PC,TIMON        ;TURN TIMER ON
2784 010206 010702                   MOV   PC,R2            ;SET RETURN PC FROM TIMER
2785 010210 032711 000020          BIT   #SDWN,(R1)       ;BRANCH WHEN SDWN SETS
2786 010214 001002                   BNE   3$

```

```

2787 010216 000163 004340          JMP      TIMER(R3)          ;GO TO TIMER & RETURN VIA R2
2788
2789 010222 004737 005126          3$:     JSR      PC, WAITRDY
2790 010226 102403                    BVS     99$
2791 010230 004737 004376          JSR     PC, TIMOK
2792 010234 000401                    BR      100$
2793
2794 010236 104400                    99$:    HLT
2795 010240 104000                    100$:   SCOPE          ;REPORT ERROR
2796
2797
2798
2799 010242 112737 000010 001122  ;TEST 010 - READ SETTLEDOWN
2800 010250 012702 010326          ;THIS TEST MEASURES TIME FROM 'SDWN'=1 TO 'SDWN'=0.
2801 010254 004737 005406          TST010: MOVB    #10, #TSTNUM      ;SET TEST #10
2802 010260 102436                    MOV     #4$, R2              ;SET RETURN PC FROM TIMER
2803 010262 004737 005332          JSR     PC, WRT.BK          ;WRITE A RECORD & BACK SPACE
2804 010266 005215                    BVS     99$
2805
2806
2807 010270 105711                    1$:     TSTB    (R1)          ;WAIT FOR READY
2808 010272 100404                    BMI     2$                  ;BRANCH WHEN SET
2809 010274 032711 040000          BIT     #ERR, (R1)         ;CHECK ERROR BIT
2810 010300 001026                    BNE     99$
2811 010302 000772                    BR      1$
2812
2813 010304 032711 000020          2$:     BIT     #SDWN, (R1)   ;WAIT FOR ASSERTION OF 'SDWN'
2814 010310 001004                    BNE     3$
2815 010312 032711 040000          BIT     #ERR, (R1)         ;MONITOR ERROR BIT
2816 010316 001017                    BNE     99$
2817 010320 000771                    BR      2$
2818
2819 010322
2820 010322 004737 004250          3$:     JSR     PC, TIMON
2821 010326 032765 000020 000012  4$:     BIT     #SDWN, DS(R5)   ;TURN TIMER ON
2822 010334 001402                    BEQ     5$                  ;WAIT FOR NEGATION OF SDWN
2823 010336 000163 004340          JMP     TIMER(R3)          ;GO TO TIMER & RETURN VIA R2
2824
2825 010342 004737 005126          5$:     JSR     PC, WAITRDY
2826 010346 102403                    BVS     99$
2827 010350 004737 004376          JSR     PC, TIMOK
2828 010354 000401                    BR      100$
2829
2830 010356 104400                    99$:    HLT
2831 010360 104000                    100$:   SCOPE
2832
2833
2834
2835 010362 112737 000011 001122  ;TEST 011-READ REVERSE START
2836 010370 012702 010426          ;THIS TEST MEASURES TIME FROM 'GO'=1 TO 'ACCL'=0.
2837 010374 004737 005314          TST011: MOVB    #11, #TSTNUM   ;SET RETURN PC FROM TIMER
2838 010400 005215                    MOV     #1$, R2              ;WRITE A RECORD
2839 010402 004737 005126          JSR     PC, WRITE          ;SET 'GO' BIT
2840 010406 102422                    INC     (R5)
2841 010410 004737 004634          JSR     PC, WAITRDY
2842 010414 004737 005350          JSR     PC, DELAY          ;WAIT FOR TAPE MOTION TO STOP
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
    
```

```

2843 010420 004737 004250      JSR    PC,TIMON      ;TURN TIMER ON
2844 010424 005215              INC    (R5)         ;SET 'GO' BIT
2845
2846 010426 005765 000032      1$:   TST    TC(R5)      ;BRANCH WHEN 'ACCL' = 0
2847 010432 100002              BPL    2$
2848 010434 000163 004340      JMP    TIMER(R3)    ;GO TO TIMER & RETURN VIA R2
2849
2850 010440 004737 005126      2$:   JSR    PC,WAITRDY   ;
2851 010444 102403              BVS    99$         ;BRANCH IF ERROR
2852 010446 004737 004376      JSR    PC,TIMOK
2853 010452 000401              BR     100$
2854
2855 010454 104400      99$:   HLT
2856 010456 104000      100$:  SCOPE
2857
2858 ;TEST 012-READ REVERSE SHUTDOWN
2859 ;THIS TEST MEASURES TIME FROM 'FC REG' = FRAME COUNT TO 'SDWN'=1.
2860 010460 112737 000012 001122  †ST012: MOVB    #12,‡TSTNUM
2861 010466 012702 010536      MOV    #35,R2      ;SET RETURN PC FROM TIMER
2862 010472 004737 005314      JSR    PC,WRITE     ;WRITE A RECORD
2863 010476 005215              INC    (R5)         ;SET 'GO' BIT
2864 010500 004737 005126      JSR    PC,WAITRDY
2865 010504 102427              BVS    99$
2866 010506 004737 005350      JSR    PC,REVRD
2867 010512 005215              INC    (R5)         ;SET 'GO' BIT
2868
2869 010514 022710 000400      1$:   CMP    #-FRMNT,(R0) ;BRANCH WHEN FRAME COUNT
2870 010520 001404              BEQ    2$          ;= # OF RECORD WRITTEN
2871 010522 032711 040000      BIT    #ERR,(R1)   ;MONITOR ERROR BIT IN 'DS' REG
2872 010526 001016              BNE    99$
2873 010530 000771              BR     1$
2874
2875 010532              2$:
2876 010532 004737 004250      JSR    PC,TIMON      ;TURN TIMER ON
2877 010536 032711 000020      3$:   BIT    #SDWN,(R1) ;BRANCH WHEN SDWN SETS
2878 010542 001002              BNE    4$
2879 010544 000163 004340      JMP    TIMER(R3)    ;GO TO TIMER & RETURN VIA R2
2880
2881 010550 004737 005126      4$:   JSR    PC,WAITRDY   ;WAIT FOR READY
2882 010554 102403              BVS    99$
2883 010556 004737 004376      JSR    PC,TIMOK
2884 010562 000401              BR     100$
2885
2886 010564 104400      99$:   HLT
2887 010566 104000      100$:  SCOPE
2888
2889 ;TEST 013-READ REVERSE SETTLEDOWN
2890 ;THIS TEST MEASURES TIME FROM 'SDWN'=1 TO 'SDWN'=0.
2891 010570 112737 000013 001122  †ST013: MOVB    #13,‡TSTNUM
2892 010576 012702 010662      MOV    #45,R2      ;SET RETURN PC FROM TIMER
2893 010602 004737 005314      JSR    PC,WRITE     ;WRITE A RECORD
2894 010606 005215              INC    (R5)         ;SET 'GO' BIT
2895 010610 004737 005126      JSR    PC,WAITRDY
2896 010614 102435              BVS    99$
2897 010616 004737 005350      JSR    PC,REVRD
2898 010622 005215              INC    (R5)         ;SET 'GO' BIT

```

```

2899
2900 010624 105711 1S: TSTB (R1) ;BRANCH WHEN
2901 010626 100404 BMI 2S ;READY SETS
2902 010630 032711 040000 BIT #ERR,(R1)
2903 010634 001025 BNE 99S
2904 010636 000772 BR 1S
2905
2906 010640 032711 000020 2S: BIT #SDWN,(R1)
2907 010644 001004 BNE 3S
2908 010646 032711 040000 BIT #ERR,(R1)
2909 010652 001016 BNE 99S
2910 010654 000771 BR 2S
2911
2912 010656 3S:
2913 010656 004737 004250 JSR PC,TIMON ;TURN TIMER ON
2914 010662 032711 000020 4S: BIT #SDWN,(R1) ;BRANCH WHEN SDWN = 0
2915 010666 001402 BEQ 5S
2916 010670 000163 004340 JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2
2917
2918 010674 004737 005126 5S: JSR PC,WAITRDY ;WAIT FOR READY
2919 010700 102403 BVS 99S
2920 010702 004737 004376 JSR PC,TIMOK
2921 010706 000401 BR 100S
2922
2923 010710 104400 99S: HLT
2924 010712 104000 100S: SCOPE
2925
2926 ;REWIND DRIVE
2927 010714 A:
2928 010714 004737 005232 JSR PC,.REWIND ;REWIND SLAVE
2929 010720 102401 BVS 99S ;BRANCH IF ERROR ON REWIND
2930 010722 102002 BVC 100S
2931 010724 104400 99S: HLT
2932 010726 000772 BR A
2933 010730 100S:
2934
2935 ;TEST 014-TURN AROUND DELAY (FORWARD-REVERSE)
2936 ;THIS TEST MEASURES TIME FROM 'GO'=1 (READ REVERSE) TO 'ACCL'=0
2937 010730 112737 000014 001122 TST014: MOVB #14,20TSTNUM
2938 010736 012702 010770 MOV #2S,R2 ;SET RETURN PC FROM TIMER
2939 010742 004737 005314 JSR PC,WRITE ;WRITE A RECORD
2940 010746 005215 INC (R5) ;SET 'GO' BIT
2941 010750 004737 005126 JSR PC,WAITRDY
2942 010754 102420 BVS 99S
2943
2944 010756 004737 005350 1S: JSR PC,REVRD ;READ THE RECORD (REVERSE)
2945 010762 004737 004250 JSR PC,TIMON ;TURN TIMER ON
2946 010766 005215 INC (R5) ;SET 'GO' BIT
2947
2948 010770 005765 000032 2S: TST TC(R5) ;WAIT FOR 'ACCL' = 0
2949 010774 100002 BPL 3S
2950 010776 000163 004340 JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2
2951
2952 011002 004737 005126 3S: JSR PC,WAITRDY
2953 011006 102403 BVS 99S
2954 011010 004737 004376 JSR PC,TIMOK
    
```



```

2955 011014 000401          BR      100$
2956 011016 104400          99$:  HLT
2957 011020 104000          100$: SCOPE
2958
2959
2960          ;TEST 015- TURN AROUND DELAY (REVERSE-FORWARD)
2961          ;THIS TEST MEASURES TIME FROM 'GO'=1 (READ) TO 'ACCL'=0.
2962 011022 112737 000015 001122 †ST015: MOVB  #15,2†STNUM
2963 011030 012702 011076          MOV  #25,R2          ;SET RETURN PC FROM TIMER
2964 011034 004737 005314          JSR  PC,WRITE        ;WRITE A RECORD
2965 011040 005215          INC  (R5)           ;SET 'GO' BIT
2966 011042 004737 005126          JSR  PC,WAITRDY     ;WAIT FOR READY
2967 011046 102426          BVS  99$
2968 011050 004737 005350          JSR  PC,REVRD       ;READ A RECORD IN THE
2969 011054 005215          INC  (R5)           ;SET 'GO' BIT
2970
2971 011056 004737 005126          JSR  PC,WAITRDY
2972 011062 102420          BVS  99$
2973
2974 011064 004737 005332 1$:  JSR  PC,READ        ;READ RECORD FORWARD
2975 011070 004737 004250          JSR  PC,TIMON       ;TURN TIMER ON
2976 011074 005215          INC  (R5)           ;SET 'GO' BIT
2977
2978 011076 005765 000032 2$:  TST  TC(R5)         ;WAIT FOR 'ACCL' = 0
2979 011102 100002          BPL  3$
2980 011104 000163 004340          JMP  TIMER(R3)      ;GO TO TIMER & RETURN VIA R2
2981
2982 011110 004737 005126 3$:  JSR  PC,WAITRDY
2983 011114 102403          BVS  99$
2984 011116 004737 004376          JSR  PC,TIMOK
2985 011122 000401          BR      100$
2986
2987 011124 104400          99$:  HLT
2988 011126 104000          100$: SCOPE
2989
2990          ;TEST 016-GAP SIZE (STOP HALF)
2991 011130 112737 000016 001122 †ST016: MOVB  #16,2†STNUM
2992 011136 012702 011174          MOV  #15,R2          ;SET RETURN PC FROM TIMER
2993 011142 004737 005314          JSR  PC,WRITE        ;WRITE A RECORD
2994 011146 005215          INC  (R5)           ;SET 'GO' BIT
2995 011150 004737 005126          JSR  PC,WAITRDY
2996 011154 102421          BVS  99$
2997 011156 004737 004634          JSR  PC,DELAY       ;DELAY 350 MS
2998 011162 004737 005350          JSR  PC,REVRD       ;READ REVERSE RECORD
2999 011166 004737 004250          JSR  PC,TIMON       ;TURN TIMER ON
3000 011172 005215          INC  (R5)           ;SET 'GO' BIT
3001
3002 011174 005710 1$:  TST  (R0)         ;WAIT FOR FRAME COUNT > 0
3003 011176 001002          BNE  2$
3004 011200 000163 004340          JMP  TIMER(R3)      ;GO TO TIMER & RETURN VIA R2
3005
3006 011204 004737 005126 2$:  JSR  PC,WAITRDY     ;WAIT FOR READY BIT TO SET
3007 011210 102403          BVS  99$
3008 011212 004737 004376          JSR  PC,TIMOK       ;CHECK TIME
3009 011216 000401          BR      100$
3010

```

```

3011 011220 104400          99$: HLT
3012 011222 104000          100$: SCOPE
3013
3014
3015 011224 112737 000017 001122 ;TEST 017-GAP SIZE (START HALF)
3016 011232 012702 011304          †TST017: MOVB #17,2#TSTNUM
3017 011236 004737 005314          MOV #18,R2 ;SET RETURN PC FROM TIMER
3018 011242 005215          JSR PC,WRITE ;WRITE A RECORD
3019 011244 004737 005126          INC (R5) ;SET 'GO' BIT
3020 011250 102427          JSR PC,WAITRDY ;WAIT FOR READY
3021 011252 004737 005350          BVS 99$
3022 011256 005215          JSR PC,REVRD ;READ REVERSE THE RECORD
3023 011260 004737 005126          INC (R5) ;SET 'GO' BIT
3024 011264 102421          JSR PC,WAITRDY ;WAIT FOR READY
3025 011266 004737 004634          BVS 99$ ;BRANCH ON ERROR
3026 011272 004737 005332          JSR PC,DELAY ;WAIT FOR TAPE MOTION TO STOP
3027 011276 004737 004250          JSR PC,READ ;READ RECORD
3028 011302 005215          JSR PC,TIMON ;TURN TIMER ON
3029          INC (R5) ;SET 'GO' BIT
3030 011304 005710          1$: TST (R0) ;WAIT FOR FRAME COUNT > 0
3031 011306 001002          BNE 2$
3032 011310 000163 004340          JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2
3033
3034 011314 004737 005126          2$: JSR PC,WAITRDY ;WAIT FOR READY
3035 011320 102403          BVS 99$
3036 011322 004737 004376          JSR PC,TIMOK ;CHECK TIME
3037 011326 000401          BR 100$
3038
3039 011330 104400          99$: HLT
3040 011332 104000          100$: SCOPE
3041
3042
3043          ;TEST 020- GAP SIZE (INTERRECORD)
3044          ;THIS TEST MEASURES TIME FROM 'GO'=1 TO 'FC REG' >0.
3045 011334 112737 000020 001122 †TST020: MOVB #20,2#TSTNUM
3046 011342 012702 011424          MOV #18,R2 ;SET RETURN PC FROM TIMER
3047 011346 004737 005314          JSR PC,WRITE ;WRITE A RECORD
3048 011352 005215          INC (R5) ;SET 'GO' BIT
3049 011354 004737 005126          JSR PC,WAITRDY ;WAIT FOR READY
3050 011360 102433          BVS 99$
3051 011362 004737 005314          JSR PC,WRITE ;WRITE SECOND RECORD
3052 011366 005215          INC (R5) ;SET 'GO' BIT
3053 011370 004737 005126          JSR PC,WAITRDY ;WAIT FOR READY
3054 011374 102425          BVS 99$
3055 011376 004737 005350          JSR PC,REVRD ;READ REVERSE SECOND RECORD
3056 011402 005215          INC (R5) ;SET 'GO' BIT
3057 011404 004737 005126          JSR PC,WAITRDY ;WAIT FOR READY
3058 011410 102417          BVS 99$
3059 011412 004737 005350          JSR PC,REVRD ;READ REVERSE FIRST RECORD
3060 011416 004737 004250          JSR PC,TIMON ;TURN TIMER ON
3061          INC (R5) ;SET 'GO' BIT
3062 011424 005710          1$: TST (R0) ;WAIT FOR FRAME COUNT > 0
3063 011426 001002          BNE 2$
3064 011430 000163 004340          JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2
3065
3066 011434 004737 005126          2$: JSR PC,WAITRDY ;WAIT FOR READY

```

3067 011440 102403  
 3068 011442 004737 004376  
 3069 011446 000401  
 3070  
 3071 011450 104400  
 3072 011452 104000  
 3073  
 3074  
 3075  
 3076  
 3077  
 3078  
 3079  
 3080  
 3081  
 3082  
 3083  
 3084  
 3085

BVS 99\$  
 JSR PC,TIMOK  
 BR 100\$  
 99\$: HLT  
 100\$: SCOPE

: TEST 021- GAP CONSISTANCY  
 : THIS TEST MEASURES TIME FROM 'GO'=1 TO 'FC REG' > 0.  
 : THE TEST REWINDS THE TAPE, WRITES 17 RECORDS WITH A DELAY FROM 1-16 MS  
 : BETWEEN EACH WRITE COMMAND. AFTER THE 17. RECORDS ARE WRITTEN THE  
 : PROGRAM READ REVERSES 16 RECORDS. AT THIS POINT THE TAPE IS STOPPED BE-  
 : TWEEN THE FIRST AND SECOND RECORD. A READ COMMAND IS EXECUTED TO READ  
 : THE 16 RECORDS WITH THE TIME BETWEEN GO=1 TO FC > 0 STORED IN 'GAPTBL'  
 : FOR EACH RECORD READ. AFTER 16 RECORDS HAVE BEEN READ THE TIME IS VER-  
 : IFIED FOR EACH READ. AFTER ALL RECORD TIMES ARE VERIFIED THEY ARE AVER-  
 : AGED AND PLACED IN THE 'ATIMTBL' (BY SCOPE). THE ABOVE PROCESS IS RE-  
 : PEATED FOR EACH ITERATION.

3086 011454 112737 000021 001122  
 3087 011462 012702 011620  
 3088 011466 004737 005232  
 3089 011472 102530  
 3090 011474 005037 001114  
 3091 011500 012700 000021  
 3092 011504 004737 005314  
 3093 011510 005215  
 3094 011512 004737 005126  
 3095 011516 102516  
 3096 011520 004737 004664  
 3097 011524 062737 000022 001114  
 3098 011532 005300  
 3099 011534 001363  
 3100  
 3101 011536 012700 000021  
 3102 011542 004737 005350  
 3103 011546 005215  
 3104 011550 004737 005126  
 3105 011554 102477  
 3106 011556 005300  
 3107 011560 001370  
 3108  
 3109 011562 012700 000020  
 3110 011566 012701 001054  
 3111 011572 004737 005332  
 3112 011576 005215  
 3113  
 3114 011600 004737 005126  
 3115 011604 102463  
 3116 011606 004737 005332  
 3117 011612 004737 004250  
 3118 011616 005215  
 3119  
 3120 011620 005765 000006  
 3121 011624 001002  
 3122 011626 000163 004340

TST021: MOV #21, #TSTNUM  
 MOV #45, R2 ; SET RETURN PC FROM TIMER  
 JSR PC, REWIND ; REWIND SLAVE  
 BVS 99\$ ; BRANCH IF ERROR ON REWIND  
 CLR DELTIM ; CLEAR VARIABLE DELAY TIME  
 MOV #17, R0 ; SET # OF RECORDS TO WRITE  
 1\$: JSR PC, WRITE ; WRITE 17. RECORDS  
 INC (R5) ; SET 'GO' BIT  
 JSR PC, WAITRDY ; WAIT FOR READY  
 BVS 99\$  
 JSR PC, DELAYV ; DELAY BEFORE WRITING NEXT REC.  
 ADD #18, DELTIM ; SET NEXT DELAY TIME  
 DEC R0 ; DECREMENT RECORDS WRITTEN COUNT  
 BNE 1\$  
 2\$: MOV #17, R0 ; SET # OF RECS. TO REVERSE READ  
 JSR PC, REVRD ; REVERSE READ 17. RECORDS  
 INC (R5) ; SET 'GO' BIT  
 JSR PC, WAITRDY ; WAIT FOR READY  
 BVS 99\$  
 DEC R0 ; DECREMENT RECORD COUNT  
 BNE 2\$  
 3\$: MOV #16, R0 ; SET # OF RECORDS TO READ  
 MOV #GAPTBL, R1 ; SET PTR TO GAP TABLE FOR TEST  
 JSR PC, READ ; READ A RECORD  
 INC (R5) ; SET 'GO' BIT  
 JSR PC, WAITRDY ; WAIT FOR READY  
 BVS 99\$  
 JSR PC, READ ; READ NEXT RECORD  
 JSR PC, TIMON ; TURN TIMER ON  
 INC (R5) ; SET 'GO' BIT  
 4\$: TST FC(R5) ; WAIT FOR FRAME COUNT > 0  
 BNE 5\$  
 JMP TIMER(R3) ; GO TO TIMER & RETURN VIA R2

```

3123
3124 011632 004737 005126 5$: JSR PC, WAITRDY ;WAIT FOR READY
3125 011636 102446 BVS 99$
3126 011640 010421 MOV R4, (R1)+ ;STORE TIME IN GAP TBL
3127 011642 005300 DEC R0 ;DECREMENT # OF RECORDS READ
3128 011644 001355 BNE 3$
3129
3130 011646 105037 001120 CLR B ;SET GAP # 0
3131 011652 012700 000020 MOV #16, R0
3132 011656 012701 001054 MOV #GAP TBL, R1
3133
3134 011662 012104 6$: MOV (R1)+, R4 ;GET GAP TICK COUNT
3135 011664 004737 004506 JSR PC, GAP OK ;CHECK TIME
3136 011670 105237 001120 INCB #GAP ;INCREMENT GAP #
3137 011674 122737 000020 001120 CMPB #16., #GAP ;BRANCH IF ALL GAPS NOT CHECKED
3138 011702 001367 BNE 6$
3139
3140 011704 012700 000020 MOV #16, R0 ;SETUP TO AVERAGE GAP SIZES
3141 011710 012701 001054 MOV #GAP TBL, R1 ;SET PTR TO TABLE
3142 011714 005002 CLR R2 ;CLEAR 'SUM' REGISTERS
3143 011716 005003 CLR R3
3144 011720 062102 7$: ADD (R1)+, R2 ;ADD ALL GAP SIZES TOGETHER
3145 011722 005503 ADC R3
3146 011724 005300 DEC R0
3147 011726 001374 BNE 7$
3148 011730 012700 000004 MOV #4, R0 ;NOW DIVIDE BY 16.
3149 011734 006203 8$: ASR R3 ;BY SHIFTING 4 PLACES RIGHT
3150 011736 006002 ROR R2
3151 011740 005300 DEC R0
3152 011742 001374 BNE 8$
3153 011744 010204 MOV R2, R4 ;MOVE AVERAGED TIMES TO R4
3154 011746 004737 004376 JSR PC, TIMOK ;CHECK AVERAGED TIMES
3155 011752 000401 BP 100$
3156
3157 011754 104400 99$: HLT
3158 011756 104000 100$: SCOPE
3159
3160
3161
3162 ;TEST 022-DATA TIME (800BPI)
3163 ;THIS TEST MEASURES THE TIME FROM FC REG >-6400 TO 'RDY' = 1.
3164 011760 112737 000022 001122 TST02: MOVB #022, #TSTNUM
3165 011766 012702 012046 MOV #3$, R2 ;SET RETURN PC FROM TIMER
3166 011772 004737 005232 JSR PC, REWIND ;REWIND SLAVE
3167 011776 102442 BVS 99$ ;BRANCH IF ERROR ON REWIND
3168 012000 052765 001700 000032 BIS #NORM11, TC(R5) ;SET 800 BPI
3169 012006 004337 005450 JSR R3, TCMO ;WRITE 3200. WORD RECORD
3170 012012 015730 .WORD WTBUF
3171 012014 171600 .WORD -3200.
3172 012016 163400 .WORD -6400.
3173 012020 000060 .WORD WFD
3174 012022 005215 INC (R5) ;SET 'GO' BIT
3175
3176 012024 022710 163400 1$: CMP #-6400., (R0) ;WAIT FOR WRITING TO START
3177 012030 001004 BNE 2$
3178 012032 032711 040000 BIT #ERR, (R1) ;MONITOR ERROR BIT

```

```

3179 012036 001022          BNE 99$
3180 012040 000771          BR 1$
3181
3182 012042          2$:
3183 012042 004737 004250      JSR PC,TIMON          ;TURN TIMER ON
3184 012046 105711          TSTB (R1)            ;BRANCH WHEN READY SETS
3185 012050 100402          BMI 4$
3186 012052 000163 004340      JMP TIMER(R3)        ;GO TO TIMER & RETURN VIA R2
3187
3188 012056 012700 000003      4$: MOV #3,R0          ;SET SHIFT COUNT
3189 012062 006204          5$: ASR R4
3190 012064 005300          DEC R0
3191 012066 001375          BNE 5$
3192 012070 004737 005126      JSR PC,WAITRDY
3193 012074 102403          BVS 99$
3194 012076 004737 004376      JSR PC,TIMOK         ;CHECK TIME
3195 012102 000401          BR 100$
3196
3197 012104 104400          99$: HLT
3198 012106 104000          100$: SCOPE
3199
3200          ;TEST 023-DATA TIME (1600BPI)
3201          ;THIS TEST MEASURES THE TIME FROM FC REG >-6400 TO 'RDY' = 1.
3202 012110 112737 000023 001122  TST023: MOVB #023,2#TSTNUM
3203 012116 012702 012204          MOV #3$,R2          ;SET RETURN PC FROM TIMER
3204 012122 004737 005232          JSR PC,REWIND       ;REWIND SLAVE
3205 012126 102442          BVS 99$             ;BRANCH IF ERROR ON REWIND
3206 012130 042765 003700 000032      BIC #3700,TC(R5)    ;CLEAR CURRENT DENSITY
3207 012136 052765 002300 000032      BIS #PE1600,TC(R5) ;SET 1600 BPI
3208 012144 004337 005450          JSR R3,TMCMO        ;WRITE 3200. WORD RECORD
3209 012150 015730          .WORD WTBUF
3210 012152 171600          .WORD -3200.
3211 012154 163400          .WORD -6400.
3212 012156 000060          .WORD WFWO
3213 012160 005215          INC (R5)           ;SET 'GO' BIT
3214
3215 012162 022710 163400          1$: CMP #-6400.,(R0)  ;BRANCH WHEN WRITING STARTS
3216 012166 001004          BNE 2$
3217 012170 032711 040000          BIT #ERR,(R1)      ;MONITOR ERROR BIT
3218 012174 001017          BNE 99$
3219 012176 000771          BR 1$
3220
3221 012200          2$:
3222 012200 004737 004250      JSR PC,TIMON          ;TURN TIMER ON
3223 012204 105711          3$: TSTB (R1)        ;BRANCH WHEN READY SETS
3224 012206 100402          BMI 4$
3225 012210 000163 004340      JMP TIMER(R3)        ;GO TO TIMER & RETURN VIA R2
3226
3227 012214 006204          4$: ASR R4          ;DIVIDE TIME BY 4
3228 012216 006204          ASR R4
3229 012220 004737 005126      JSR PC,WAITRDY
3230 012224 102403          BVS 99$
3231 012226 004737 004376      JSR PC,TIMOK         ;CHECK TIME
3232 012232 000401          BR 100$
3233
3234 012234 104400          99$: HLT

```

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER  
 DZTEEA.P11 31-MAR-77 00:00

MACY11 30(1046) 27-JUN-77 17:03 PAGE 72  
 START OF TESTS

```

3235 012236 104000 100$: SCOPE
3236
3237
3238
3239 012240 112737 000024 001122 ;TEST 024-ERASE
;THIS TEST MEASURES TIME FROM 'GO'=1 TO 'RDY'=1.
3240 012246 012702 012330 TST024: MOVB #24,2#TSTNUM
3241 012252 004737 005232 MOV #25,R2 ;SET RETURN PC FROM TIMER
3242 012256 102436 JSR PC,REWIND ;REWIND SLAVE
3243 012260 004737 005070 BVS 99$ ;BRANCH IF ERROR ON REWIND
3244 012264 004737 005314 JSR PC,RHINIT ;SET NRZ
3245 012270 005215 INC (R5) ;WRITE A RECORD
3246 012272 004737 005126 JSR PC,WAITRDY ;SET 'GO' BIT
3247 012276 102426 BVS 99$
3248 012300 012737 012306 001002 MOV #15,2#SCPADR
3249 012306 004337 005450 1$: JSR R3,2#TMCMD
3250 012312 000000 .WORD 0
3251 012314 000000 .WORD 0
3252 012316 000000 .WORD 0
3253 012320 000024 .WORD ERASE
3254 012322 004737 004250 JSR PC,TIMON ;TURN TIMER ON
3255 012326 005215 INC (R5) ;SET 'GO' BIT
3256
3257 012330 105711 2$: TSTB (R1) ;BRANCH WHEN READY SETS
3258 012332 100402 BMI 3$
3259 012334 000163 004340 JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2
3260
3261 012340 004737 005126 3$: JSR PC,WAITRDY
3262 012344 102403 BVS 99$
3263 012346 004737 004376 JSR PC,TIMOK
3264 012352 000401 BR 100$
3265
3266 012354 104400 99$: HLT
3267 012356 104000 100$: SCOPE
3268
3269
3270 ;TEST 025 TAPE MARK
;THIS TEST MEASURES TIME FROM 'GO'=1 TO 'RDY'=1.
3271 012360 112737 000025 001122 TST025: MOVB #25,2#TSTNUM
3272 012366 012702 012430 MOV #15,R2 ;SET RETURN PC FROM TIMER
3273 012372 004737 005314 JSR PC,WRITE ;WRITE A RECORD
3274 012376 005215 INC (R5) ;SET 'GO' BIT
3275 012400 004737 005126 JSR PC,WAITRDY
3276 012404 102423 BVS 99$
3277 012406 004337 005450 JSR R3,2#TMCMD
3278 012412 000000 .WORD 0
3279 012414 000000 .WORD 0
3280 012416 000000 .WORD 0
3281 012420 000026 .WORD WFMK
3282 012422 004737 004250 JSR PC,TIMON ;TURN TIMER ON
3283 012426 005215 INC (R5) ;SET 'GO' BIT
3284
3285 012430 105711 1$: TSTB (R1) ;BRANCH WHEN READY SETS
3286 012432 100402 BMI 2$
3287 012434 000163 004340 JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2
3288
3289 012440 004737 005126 2$: JSR PC,WAITRDY
3290 012444 102403 BVS 99$

```

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER  
DZTEEA.P11 31-MAR-77 00:00

MACY11 30(1046) 27-JUN-77 17:03 PAGE 73  
START OF TESTS

3291 012446 004737 004376  
3292 012452 000401  
3293  
3294 012454 104400  
3295 012456  
3296 012456 004737 005232  
3297 012462 102774  
3298 012464 104000  
3299

JSR PC TIMOK  
BR 100\$  
99\$: HLT  
100\$: JSR PC .REWIND  
BVS 99\$  
SCOPE

;REWIND SLAVE  
;BRANCH IF ERROR ON REWIND

|      |        |        |        |        |         |              |              |  |                                 |
|------|--------|--------|--------|--------|---------|--------------|--------------|--|---------------------------------|
| 3300 | 012466 | 032777 | 002000 | 166304 | FINISH: | BIT          | #SW10, #SWR  |  | ; DO NOT SPACE PAPER            |
| 3301 | 012474 | 001011 |        |        |         | BNE          | 2S           |  | ; IF USER SELECTED NO OUTPUT    |
| 3302 | 012476 | 005737 | 005716 |        |         | TST          | CHNFLG       |  | ; OR IF IN CHAIN MODE           |
| 3303 | 012502 | 001006 |        |        |         | BNE          | 2S           |  |                                 |
| 3304 | 012504 | 012700 | 000012 |        |         | MOV          | #10., RO     |  | ; SET LINE FEED COUNT           |
| 3305 | 012510 | 000004 | 001374 |        | 1S:     | TYPE, CRLF   |              |  |                                 |
| 3306 | 012514 | 005300 |        |        |         | DEC          | RO           |  |                                 |
| 3307 | 012516 | 001374 |        |        |         | BNE          | 1S           |  |                                 |
| 3308 |        |        |        |        |         |              |              |  |                                 |
| 3309 |        |        |        |        |         |              |              |  |                                 |
| 3310 | 012520 | 105237 | 001005 |        | 2S:     | INCB         | @SLVNUM      |  | ; SET NEXT SLAVE #              |
| 3311 | 012524 | 005237 | 001006 |        |         | INC          | @SLVPTR      |  | ; AND ITS POINTER               |
| 3312 | 012530 | 122737 | 000010 | 001005 |         | CMPB         | #8., @SLVNUM |  | ; BRANCH IF LAST SLAVE (7)      |
| 3313 | 012536 | 001402 |        |        |         | BEQ          | 3S           |  |                                 |
| 3314 | 012540 | 000137 | 006750 |        |         | JMP          | @BEGIN       |  | ; BEGIN TEST ON NEXT SLAVE      |
| 3315 | 012544 | 105037 | 001005 |        | 3S:     | CLRB         | @SLVNUM      |  | ; SET SLAVE #0                  |
| 3316 | 012550 | 105237 | 001004 |        |         | INCB         | @DRVNUM      |  | ; AND INCREMENT DRIVE #         |
| 3317 | 012554 | 122737 | 000010 | 001004 |         | CMPB         | #8., @DRVNUM |  | ; AND CHECK IF LAST DRIVE       |
| 3318 | 012562 | 001402 |        |        |         | BEQ          | END          |  |                                 |
| 3319 | 012564 | 000137 | 006750 |        |         | JMP          | @BEGIN       |  |                                 |
| 3320 |        |        |        |        |         |              |              |  |                                 |
| 3321 | 012570 | 105737 | 001125 |        | END:    | TSTB         | @UNTFND      |  | ; BRANCH IF A UNIT WAS FOUND    |
| 3322 | 012574 | 001004 |        |        |         | BNE          | 1S           |  |                                 |
| 3323 | 012576 | 000004 | 014147 |        |         | TYPE, E.UNIT |              |  |                                 |
| 3324 | 012602 | 000137 | 005620 |        |         | JMP          | @INIT        |  |                                 |
| 3325 | 012606 | 105237 | 001127 |        | 1S:     | INCB         | @PSCNT       |  | ; INCREMENT PASS COUNT          |
| 3326 | 012612 | 000004 | 013406 |        |         | TYPE, M.EOP  |              |  |                                 |
| 3327 | 012616 | 113702 | 001127 |        |         | MOVB         | @PSCNT, R2   |  | ; GET PASSCOUNT                 |
| 3328 | 012622 | 004737 | 002426 |        |         | JSR          | PC, TYPOCT   |  | ; AND TYPE IT                   |
| 3329 | 012626 | 000004 | 001374 |        |         | TYPE, CRLF   |              |  |                                 |
| 3330 | 012632 | 013700 | 000042 |        |         | MOV          | @42, RO      |  | ; GET ACT11 RETURN ADDRESS      |
| 3331 | 012636 | 001405 |        |        |         | BEQ          | HERE         |  | ; BRANCH IF NOT ACT11           |
| 3332 | 012640 | 000005 |        |        |         | RESET        |              |  |                                 |
| 3333 | 012642 | 004710 |        |        | SENDAD: | JSR          | PC, (RO)     |  |                                 |
| 3334 | 012644 | 000240 |        |        |         | NOP          |              |  |                                 |
| 3335 | 012646 | 000240 |        |        |         | NOP          |              |  |                                 |
| 3336 | 012650 | 000240 |        |        |         | NOP          |              |  |                                 |
| 3337 | 012652 | 000240 |        |        | HERE:   | NOP          |              |  |                                 |
| 3338 | 012654 | 005737 | 005716 |        |         | TST          | CHNFLG       |  | ; BRANCH IF CHAIN MODE          |
| 3339 | 012660 | 001004 |        |        |         | BNE          | 1S           |  |                                 |
| 3340 | 012662 | 032777 | 000100 | 166110 |         | BIT          | #SW06, #SWR  |  | ; BRANCH IF NOT CONTINUOUS LOOP |
| 3341 | 012670 | 001402 |        |        |         | BEQ          | 2S           |  |                                 |
| 3342 | 012672 | 000137 | 006726 |        | 1S:     | JMP          | @RSTRT       |  | ; RESTART                       |
| 3343 | 012676 | 000000 |        |        | 2S:     | HALT         |              |  |                                 |
| 3344 | 012700 | 000005 |        |        |         | RESET        |              |  |                                 |
| 3345 | 012702 | 000137 | 005620 |        |         | JMP          | @INIT        |  | ; RESTART                       |



```

3346 ;SKEW TAPE TIMING TESTS
3347 ;THE FOLLOWING TESTS REQUIRE A SPECIALLY WRITTEN 800 BPI SKEW TAPE
3348 012706 012737 012714 001002 SKENTST:MOV #TST026,#SCPADR ;SET SCOPE POINTER
3349
3350 ;TEST 026- SKEW TAPE SPEED TEST-FORWARD
3351 ;THIS TEST READS 32" OF TAPE (26400.-800. = 25600. FRAMES) THEN
3352 ;DIVIDES TIME BY 32. TO GET TIME TO READ 1" (800. FRAMES) OF TAPE.
3353 012714 112737 000026 001122 TST026: MOV #26,#TSTNUM
3354 012722 012702 013000 MOV #25,R2 ;SET RETURN PC FROM TIMER
3355 012726 004737 005232 JSR PC,REWIND ;REWIND SLAVE
3356 012732 102441 BVS 99$ ;BRANCH IF ERROR ON REWIND
3357 012734 052765 001700 000032 BIS #NORM11,TC(R5) ;SET 800 BPI
3358 012742 052765 000010 000010 BIS #BA1,CS2(R5) ;INHIBIT BUS ADDRESS INCREMENT
3359 012750 004337 005450 JSR R3,#TMCMD ;READ 32" OF TAPE-FORWARD
3360 012754 015730 .WORD R0BUF
3361 012756 177777 .WORD -1.
3362 012760 063440 10$: .WORD 26400. ;FRAME COUNT
3363 012762 000070 .WORD R0FWD
3364 012764 005215 INC (R5) ;SET 'GO' BIT
3365
3366 012766 022710 001440 1$: CMP #800.,(R0) ;WAIT FOR FIRST 800 FRAMES
3367 012772 101375 BHI 1$ ;TO BE READ
3368
3369 012774 004737 004250 JSR PC,TIMON ;TURN TIMER ON
3370 013000 023710 012760 2$: CMP #10$, (R0) ;WAIT FOR READING TO FINISH
3371 013004 103402 BLO 3$
3372 013006 000163 004340 JMP TIMER(R3) ;GO TO TIMER & RETURN VIA R2
3373
3374 013012 012700 000005 3$: MOV #5,R0 ;DIVIDE TIME BY 32.
3375 013016 006204 4$: ASR R4
3376 013020 005300 DEC R0
3377 013022 001375 BNE 4$
3378 013024 004737 005070 JSR PC,RHINIT ;INIT DRIVE
3379 013030 004737 004376 JSR PC,TIMOK ;CHECK TIME
3380 013034 000401 BR 100$
3381
3382 013036 104400 99$: HLT
3383 013040 104000 100$: SCOPE
3384
3385 ;TEST 027-SKEW TAPE SPEED TEST-REVERSE
3386 ;THIS TEST READS FORWARD 40" (32000. FRAMES) OF TAPE, THEN READS REVERSE
3387 ;32" (26400.-800. = 25600. FRAMES) OF TAPE. THE TIME IS THEN DIVIDED BY
3388 ;32. TO GET TIME TO READ 1" (800. FRAMES) OF TAPE.
3389 013042 112737 000027 001122 TST027: MOV #27,#TSTNUM
3390 013050 012702 013176 MOV #3$,R2 ;SET RETURN PC FROM TIMER
3391 013054 004737 005232 JSR PC,REWIND ;REWIND SLAVE
3392 013060 102465 BVS 99$ ;BRANCH IF ERROR ON REWIND
3393 013062 052765 001700 000032 BIS #NORM11,TC(R5)
3394 013070 052765 000010 000010 BIS #BA1,CS2(R5)
3395 013076 004337 005450 JSR R3,#TMCMD ;READ FORWARD 32000. FRAMES
3396 013102 015730 .WORD R0BUF
3397 013104 177777 .WORD -1. ;WORD COUNT
3398 013106 076400 10$: .WORD 32000. ;FRAME COUNT
3399 013110 000070 .WORD R0FWD ;READ FORWARD
3400 013112 005215 INC (R5) ;SET 'GO' BIT
3401

```

|      |        |        |        |        |       |                 |                                  |
|------|--------|--------|--------|--------|-------|-----------------|----------------------------------|
| 3402 | 013114 | 023710 | 013106 | 15:    | CMP   | 2#10\$, (R0)    |                                  |
| 3403 | 013120 | 101375 |        |        | BHI   | 15              |                                  |
| 3404 |        |        |        |        |       |                 |                                  |
| 3405 | 013122 | 004737 | 005070 |        | JSR   | PC, RHINIT      | ; INIT DRIVE                     |
| 3406 | 013126 | 004737 | 004634 |        | JSR   | PC, DELAY       | ; WAIT FOR TAPE MOTION TO STOP   |
| 3407 | 013132 | 052765 | 001700 | 000032 | BIS   | #NORM11, TC(R5) | ; SET 800 BPI                    |
| 3408 | 013140 | 052765 | 000010 | 000010 | BIS   | #BAT, CS2(R5)   | ; INHIBIT BUS ADDRESS INCREMENT  |
| 3409 | 013146 | 004337 | 005450 |        | JSR   | R3, 2#TMCMD     | ; READ REVERSE 32" OF TAPE       |
| 3410 | 013152 | 015730 |        |        | .WORD | RDBUF           | ; READ BUFFER                    |
| 3411 | 013154 | 177777 |        |        | .WORD | -1.             | ; WORD COUNT                     |
| 3412 | 013156 | 063440 |        | 115:   | .WORD | 26400.          | ; FRAME COUNT                    |
| 3413 | 013160 | 000076 |        |        | .WORD | RDRREV          | ; READ REVERSE                   |
| 3414 | 013162 | 005215 |        |        | INC   | (R5)            | ; SET 'GO' BIT                   |
| 3415 |        |        |        |        |       |                 |                                  |
| 3416 | 013164 | 022710 | 001440 | 25:    | CMP   | #800., (R0)     | ; WAIT FOR FIRST 800 FRAMES      |
| 3417 | 013170 | 101375 |        |        | BHI   | 25              | ; TO BE READ                     |
| 3418 |        |        |        |        |       |                 |                                  |
| 3419 | 013172 | 004737 | 004250 |        | JSR   | PC, TIMON       | ; TURN TIMER ON                  |
| 3420 | 013176 | 023710 | 013156 | 35:    | CMP   | 2#11\$, (R0)    | ; WAIT FOR ALL FRAMES TO BE READ |
| 3421 | 013202 | 103402 |        |        | BLO   | 45              |                                  |
| 3422 | 013204 | 000163 | 004340 |        | JMP   | TIMER(R3)       | ; GO TO TIMER & RETURN VIA R2    |
| 3423 |        |        |        |        |       |                 |                                  |
| 3424 | 013210 | 012700 | 000005 | 45:    | MOV   | #5, R0          | ; DIVIDE TIME BY 32.             |
| 3425 | 013214 | 006204 |        | 55:    | ASR   | R4              |                                  |
| 3426 | 013216 | 005300 |        |        | DEC   | R0              |                                  |
| 3427 | 013220 | 001375 |        |        | BNE   | 55              |                                  |
| 3428 | 013222 | 004737 | 005070 |        | JSR   | PC, RHINIT      |                                  |
| 3429 | 013226 | 004737 | 004376 |        | JSR   | PC, TIMOK       |                                  |
| 3430 | 013232 | 000401 |        |        | BR    | 100\$           |                                  |
| 3431 |        |        |        |        |       |                 |                                  |
| 3432 | 013234 | 104400 |        | 99\$:  | HLT   |                 |                                  |
| 3433 | 013236 |        |        | 100\$: |       |                 |                                  |
| 3434 | 013236 | 004737 | 005232 |        | JSR   | PC, .REWIND     | ; REWIND SLAVE                   |
| 3435 | 013242 | 102774 |        |        | BVS   | 99\$            | ; BRANCH IF ERROR ON REWIND      |
| 3436 | 013244 | 104000 |        |        | SCOPE |                 |                                  |
| 3437 |        |        |        |        |       |                 |                                  |
| 3438 | 013246 | 000137 | 012466 |        | JMP   | 2#FINISH        |                                  |
| 3439 |        |        |        |        |       |                 |                                  |
| 3440 |        |        |        |        |       |                 |                                  |
| 3441 |        |        |        |        |       |                 |                                  |

|      |        |        |        |        |
|------|--------|--------|--------|--------|
| 3442 |        |        |        |        |
| 3443 |        |        |        |        |
| 3444 | 013252 | 005015 | 046524 | 031460 |
| 3445 | 013260 | 052057 | 030505 | 020066 |
| 3446 | 013266 | 051104 | 053111 | 020105 |
| 3447 | 013274 | 052506 | 041516 | 044524 |
| 3448 | 013302 | 047117 | 052040 | 046511 |
| 3449 | 013310 | 051105 | 024040 | 055104 |
| 3450 | 013316 | 047524 | 026505 | 024501 |
| 3451 | 013324 | 005015 | 054524 | 042520 |
| 3452 | 013332 | 036040 | 051103 | 020076 |
| 3453 | 013340 | 047524 | 052040 | 051105 |
| 3454 | 013346 | 044515 | 040516 | 042524 |
| 3455 | 013354 | 051040 | 051505 | 047520 |
| 3456 | 013362 | 051516 | 020105 | 020046 |
| 3457 | 013370 | 041536 | 052040 | 020117 |
| 3458 | 013376 | 042522 | 052123 | 051101 |
| 3459 | 013404 | 000124 |        |        |
| 3460 | 013406 | 005015 | 047105 | 020104 |
| 3461 | 013414 | 043117 | 050040 | 051501 |
| 3462 | 013422 | 020123 | 000    |        |
| 3463 | 013425 | 015    | 044012 | 051101 |
| 3464 | 013432 | 053504 | 051101 | 020105 |
| 3465 | 013440 | 053523 | 020122 | 047111 |
| 3466 | 013446 | 052440 | 042523 | 005015 |
| 3467 | 013454 | 000    |        |        |
| 3468 | 013455 | 015    | 051012 | 046505 |
| 3469 | 013462 | 053117 | 020105 | 046524 |
| 3470 | 013470 | 050104 | 043040 | 047522 |
| 3471 | 013476 | 020115 | 042524 | 033061 |
| 3472 | 013504 | 052040 | 020117 | 042502 |
| 3473 | 013512 | 052040 | 051505 | 042524 |
| 3474 | 013520 | 000104 |        |        |
| 3475 | 013522 | 005015 | 054524 | 042520 |
| 3476 | 013530 | 043040 | 051111 | 052123 |
| 3477 | 013536 | 040440 | 042104 | 042522 |
| 3478 | 013544 | 051523 | 047440 | 020106 |
| 3479 | 013552 | 047503 | 052116 | 047522 |
| 3480 | 013560 | 046114 | 051105 | 020040 |
| 3481 | 013566 | 000    |        |        |
| 3482 | 013567 | 124    | 050131 | 020105 |
| 3483 | 013574 | 046524 | 031460 | 042040 |
| 3484 | 013602 | 044522 | 042526 | 021440 |
| 3485 | 013610 | 051447 | 052040 | 020117 |
| 3486 | 013616 | 042502 | 052040 | 051505 |
| 3487 | 013624 | 042524 | 035104 | 020040 |
| 3488 | 013632 | 046101 | 020114 | 000    |
| 3489 | 013637 | 106    | 051117 | 052040 |
| 3490 | 013644 | 030115 | 020063 | 051104 |
| 3491 | 013652 | 053111 | 020105 |        |
| 3492 | 013656 | 026460 | 052040 | 050131 |
| 3493 | 013664 | 020105 | 046123 | 053101 |
| 3494 | 013672 | 020105 | 023443 | 020123 |
| 3495 | 013700 | 047524 | 041040 | 020105 |
| 3496 | 013706 | 042524 | 052123 | 042105 |
| 3497 | 013714 | 020072 | 046101 | 020114 |

.SBTTL PROGRAM MESSAGES  
 .OPERATOR INSTRUCTIONS  
 A.NAM: .ASCII <CR><LF>'TMO3/TE16 DRIVE FUNCTION TIMER (DZTEE-A)'

.ASCIZ <CR><LF>'TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART'

M.EOP: .ASCIZ <CR><LF>'END OF PASS '

M.HSWR: .ASCIZ <CR><LF>'HARDWARE SWR IN USE'<CR><LF>

I.REM: .ASCIZ <CR><LF>'REMOVE TMDP FROM TE16 TO BE TESTED'

I.REG: .ASCIZ <CR><LF>'TYPE FIRST ADDRESS OF CONTROLLER '

I.DRVS: .ASCIZ %TYPE TMO3 DRIVE #'S TO BE TESTED: ALL %

I.SLVS: .ASCII 'FOR TMO3 DRIVE '

I.DRV: .ASCIZ %- TYPE SLAVE #'S TO BE TESTED: ALL %

|      |        |        |        |        |   |
|------|--------|--------|--------|--------|---|
| 3498 | 013722 | 000    |        |        |   |
| 3499 | 013723 | 123    | 042520 | 042105 | I.SPD: .ASCIZ 'SPEED TESTS' (YES/NO): NO '                                    |
| 3500 | 013730 | 052040 | 051505 | 051524 |   |
| 3501 | 013736 | 020077 | 054450 | 051505 |   |
| 3502 | 013744 | 047057 | 024517 | 020072 |   |
| 3503 | 013752 | 047516 | 000040 |        |   |
| 3504 | 013756 | 005015 | 047105 | 020104 | M.EOT: .ASCIZ <CR><LF>'END OF TAPE'<CR><LF>                                   |
| 3505 | 013764 | 043117 | 052040 | 050101 |   |
| 3506 | 013772 | 006505 | 000012 |        |   |
| 3507 |        |        |        |        |   |
| 3508 |        |        |        |        | .ERROR MESSAGES   |
| 3509 | 013776 | 005015 | 051124 | 050101 | E.TRP4: .ASCIZ <CR><LF>'TRAPPED TO 4'   |
| 3510 | 014004 | 042520 | 020104 | 047524 |   |
| 3511 | 014012 | 032040 | 000    |        |   |
| 3512 | 014015 | 116    | 020117 | 047503 | E.NCON: .ASCIZ 'NO CONTROLLER AT ADDRESS SPECIFIED'<CR><LF>                   |
| 3513 | 014022 | 052116 | 047522 | 046114 |   |
| 3514 | 014030 | 051105 | 040440 | 020124 |   |
| 3515 | 014036 | 042101 | 051104 | 051505 |   |
| 3516 | 014044 | 020123 | 050123 | 041505 |   |
| 3517 | 014052 | 043111 | 042511 | 006504 |   |
| 3518 | 014060 | 000012 |        |        |   |
| 3519 | 014062 | 046524 | 031460 | 042040 | E.NDRV: .ASCIZ 'TMO3 DRIVE '  |
| 3520 | 014070 | 044522 | 042526 | 000040 |   |
| 3521 | 014076 | 051104 | 053111 | 020105 | E.NSLV: .ASCII 'DRIVE '   |
| 3522 | 014104 | 020060 | 046123 | 053101 | E.DRV: .ASCII '0 SLAVE '  |
| 3523 | 014112 | 020105 |        |        |   |
| 3524 | 014114 | 020060 | 047516 | 020124 | E.NAVA: .ASCIZ '0 NOT AVAILABLE FOR TEST'<CR><LF>                             |
| 3525 | 014122 | 053101 | 044501 | 040514 |   |
| 3526 | 014130 | 046102 | 020105 | 047506 |   |
| 3527 | 014136 | 020122 | 042524 | 052123 |   |
| 3528 | 014144 | 005015 | 000    |        |   |
| 3529 | 014147 | 116    | 020117 | 046524 | E.UNIT: .ASCIZ 'NO TMO3/TE16 UNIT FOUND TO TEST'<CR><LF>                      |
| 3530 | 014154 | 031460 | 052057 | 030505 |   |
| 3531 | 014162 | 020066 | 047125 | 052111 |   |
| 3532 | 014170 | 043040 | 052517 | 042116 |   |
| 3533 | 014176 | 052040 | 020117 | 042524 |   |
| 3534 | 014204 | 052123 | 005015 | 000    |   |
| 3535 | 014211 | 123    | 043117 | 020124 | E.SFT: .ASCIZ 'SOFT ERROR (DATA)'<<CR><LF>                                    |
| 3536 | 014216 | 051105 | 047522 | 020122 |   |
| 3537 | 014224 | 042050 | 052101 | 024501 |   |
| 3538 | 014232 | 005015 | 000    |        |   |
| 3539 | 014235 | 124    | 051505 | 020124 | E.HDR: .ASCIZ 'TEST # '   |
| 3540 | 014242 | 020043 | 000    |        |   |
| 3541 | 014245 | 040    | 042504 | 044526 | E.HDR1: .ASCII ' DEVICE ERROR'<CR><LF>  |
| 3542 | 014252 | 042503 | 042440 | 051122 |   |
| 3543 | 014260 | 051117 | 005015 |        |   |
| 3544 | 014264 | 051503 | 004461 | 041527 | .ASCIZ 'CS1'<HT>'WC'<HT>'BA'<HT>'FC'<HT>'CS2'<HT>'DS'<HT>'ER'<HT>'TC'<CR><LF> |
| 3545 | 014272 | 041011 | 004501 | 041506 |   |
| 3546 | 014300 | 041411 | 031123 | 042011 |   |
| 3547 | 014306 | 004523 | 051105 | 052011 |   |
| 3548 | 014314 | 006503 | 000012 |        |   |
| 3549 | 014320 | 047440 | 052123 | 047440 | E.HDR2: .ASCIZ ' OUT OF RANGE ERROR'<CR><LF>                                  |
| 3550 | 014326 | 020106 | 040522 | 043516 |   |
| 3551 | 014334 | 020105 | 051105 | 047522 |   |
| 3552 | 014342 | 006522 | 000012 |        |   |
| 3553 | 014346 | 005015 | 044524 | 042515 | E.TIMOV: .ASCIZ <CR><LF>'TIMER OVERFLOWED'<CR><LF>                            |

|      |        |        |        |        |
|------|--------|--------|--------|--------|
| 3554 | 014354 | 020122 | 053117 | 051105 |
| 3555 | 014362 | 046106 | 053517 | 042105 |
| 3556 | 014370 | 005015 | 000    |        |
| 3557 | 014373 | 015    | 052012 | 046511 |
| 3558 | 014400 | 020105 | 054105 | 044520 |
| 3559 | 014406 | 042522 | 020104 | 040527 |
| 3560 | 014414 | 052111 | 047111 | 020107 |
| 3561 | 014422 | 047506 | 020122 | 042122 |
| 3562 | 014430 | 006531 | 000012 |        |
| 3563 | 014434 | 043440 | 050101 | 021440 |
| 3564 | 014442 | 000040 |        |        |
| 3565 |        |        |        |        |
| 3566 |        |        |        |        |
| 3567 | 014444 | 025052 | 025052 | 025052 |
| 3568 | 014452 | 025052 | 025052 | 025052 |
| 3569 | 014460 | 025052 | 025052 | 025052 |
| 3570 | 014466 | 025052 | 025052 | 025052 |
| 3571 | 014474 | 025052 | 025052 | 025052 |
| 3572 | 014502 | 025052 | 025052 | 025052 |
| 3573 | 014510 | 025052 | 025052 | 025052 |
| 3574 | 014516 | 025052 | 025052 | 025052 |
| 3575 | 014524 | 025052 | 025052 | 025052 |
| 3576 | 014532 | 025052 | 025052 | 025052 |
| 3577 | 014540 | 025052 | 025052 | 025052 |
| 3578 | 014546 | 025052 | 025052 | 025052 |
| 3579 | 014554 | 005015 | 000    |        |
| 3580 | 014557 | 052    | 052040 | 030115 |
| 3581 | 014564 | 020063 | 051104 | 053111 |
| 3582 | 014572 | 020105 | 052506 | 041516 |
| 3583 | 014600 | 044524 | 047117 | 052040 |
| 3584 | 014606 | 046511 | 051505 | 020055 |
| 3585 | 014614 | 051104 | 053111 | 020105 |
| 3586 | 014622 | 020043 |        |        |
| 3587 | 014624 | 020060 | 046123 | 053101 |
| 3588 | 014632 | 020105 | 020043 |        |
| 3589 | 014636 | 020060 | 040    |        |
| 3590 | 014641 | 071    | 041440 | 040510 |
| 3591 | 014646 | 027116 | 051440 | 051105 |
| 3592 | 014654 | 021440 | 000040 |        |
| 3593 | 014660 | 006440 | 025012 | 005015 |
| 3594 | 014666 | 020052 | 052506 | 041516 |
| 3595 | 014674 | 044524 | 047117 | 004411 |
| 3596 | 014702 | 044524 | 042515 | 051450 |
| 3597 | 014710 | 042520 | 044503 | 044506 |
| 3598 | 014716 | 040203 | 044524 | 047117 |
| 3599 | 014724 | 004451 | 044524 | 042515 |
| 3600 | 014732 | 040450 | 052103 | 040525 |
| 3601 | 014740 | 024514 | 005015 | 000    |
| 3602 |        |        |        |        |
| 3603 | 014745 | 122    | 047101 | 042507 |
| 3604 | 014752 | 036075 | 000    |        |
| 3605 | 014755 | 101    | 052103 | 040525 |
| 3606 | 014762 | 036514 | 000    |        |
| 3607 |        |        |        |        |
| 3608 |        |        |        |        |
| 3609 | 014765 | 052    | 053440 | 044522 |

E.TIMEX: .ASCIZ <CR><LF>'TIME EXPIRED WAITING FOR RDY'<CR><LF>

E.GAP: .ASCIZ ' GAP # '

:TIME DOCUMENT LINES

L.HDR1: .ASCIZ '\*\*\*\*\*'

L.HDR2: .ASCII '\* TMO3 DRIVE FUNCTION TIMES- DRIVE # '

L.DRV: .ASCII 'D SLAVE # '

L.SLV: .ASCII 'D '

L.CHAN: .ASCIZ '9 CHAN. SER # '

L.HDR3: .ASCII ' '<CR><LF>'\* '<CR><LF>  
.ASCIZ '\* FUNCTION'<HT><HT>'TIME(SPECIFICATION)'<HT>'TIME(ACTUAL)'<CR><LF>

L.RNG: .ASCIZ 'RANGE='

L.ACT: .ASCIZ 'ACTUAL='

:TEST DESCRIPTOR HEADERS

A.T001: .ASCIZ '\* WRITE FROM BOT'<HT>

|      |        |        |        |        |  |
|------|--------|--------|--------|--------|--|
| 3610 | 014772 | 042524 | 043040 | 047522 |  |
| 3611 | 015000 | 020115 | 047502 | 004524 |  |
| 3612 | 015006 | 000    |        |        |  |
| 3613 | 015007 | 052    | 053440 | 044522 | A.T002: .ASCIZ '* WRITE START'<HT><HT>       |
| 3614 | 015014 | 042524 | 051440 | 040524 |  |
| 3615 | 015022 | 052123 | 004411 | 000    |  |
| 3616 | 015027 | 053440 | 053440 | 044522 | A.T003: .ASCIZ '* WRITE SHUTDOWN'<HT>        |
| 3617 | 015034 | 042524 | 051440 | 052510 |  |
| 3618 | 015042 | 042124 | 053517 | 004516 |  |
| 3619 | 015050 | 000    |        |        |  |
| 3620 | 015051 | 052    | 053440 | 044522 | A.T004: .ASCIZ '* WRITE SETTLEDOWN'<HT>      |
| 3621 | 015056 | 042524 | 051440 | 052105 |  |
| 3622 | 015064 | 046124 | 042105 | 053517 |  |
| 3623 | 015072 | 004516 | 000    |        |  |
| 3624 | 015075 | 052    | 051040 | 040505 | A.T005: .ASCIZ '* READ FROM BOT'<HT><HT>     |
| 3625 | 015102 | 020104 | 051106 | 046517 |  |
| 3626 | 015110 | 041046 | 052117 | 004411 |  |
| 3627 | 015116 | 000    |        |        |  |
| 3628 | 015117 | 052    | 051040 | 040505 | A.T006: .ASCIZ '* READ START'<HT><HT>        |
| 3629 | 015124 | 020104 | 052123 | 051101 |  |
| 3630 | 015132 | 004524 | 000011 |        |  |
| 3631 | 015136 | 020052 | 042522 | 042101 | A.T007: .ASCIZ '* READ SHUTDOWN'<HT><HT>     |
| 3632 | 015144 | 051440 | 052510 | 042124 |  |
| 3633 | 015152 | 053517 | 004516 | 000011 |  |
| 3634 | 015160 | 020052 | 042522 | 042101 | A.T010: .ASCIZ '* READ SETTLEDOWN'<HT>       |
| 3635 | 015166 | 051440 | 052105 | 046124 |  |
| 3636 | 015174 | 042105 | 053517 | 004516 |  |
| 3637 | 015202 | 000    |        |        |  |
| 3638 | 015203 | 052    | 051040 | 040505 | A.T011: .ASCIZ '* READ REV START'<HT>        |
| 3639 | 015210 | 020104 | 042522 | 020126 |  |
| 3640 | 015216 | 052123 | 051101 | 004524 |  |
| 3641 | 015224 | 000    |        |        |  |
| 3642 | 015225 | 052    | 051040 | 040505 | A.T012: .ASCIZ '* READ REV SHUTDOWN'<HT>     |
| 3643 | 015232 | 020104 | 042522 | 020126 |  |
| 3644 | 015240 | 044123 | 052125 | 047504 |  |
| 3645 | 015246 | 047127 | 000011 |        |  |
| 3646 | 015252 | 020052 | 042522 | 042101 | A.T013: .ASCIZ '* READ REV SETTLEDOWN'<HT>   |
| 3647 | 015260 | 051040 | 053105 | 051440 |  |
| 3648 | 015266 | 052105 | 046124 | 042105 |  |
| 3649 | 015274 | 053517 | 004516 | 000    |  |
| 3650 | 015301 | 052    | 052040 | 051125 | A.T014: .ASCIZ '* TURN AROUND DELAY F-R'<HT> |
| 3651 | 015306 | 020116 | 051101 | 052517 |  |
| 3652 | 015314 | 042116 | 042040 | 046105 |  |
| 3653 | 015322 | 054501 | 043040 | 051055 |  |
| 3654 | 015330 | 000011 |        |        |  |
| 3655 | 015332 | 020052 | 052524 | 047122 | A.T015: .ASCIZ '* TURN AROUND DELAY R-F'<HT> |
| 3656 | 015340 | 040440 | 047522 | 047125 |  |
| 3657 | 015346 | 020104 | 042504 | 040514 |  |
| 3658 | 015354 | 020131 | 026522 | 004506 |  |
| 3659 | 015362 | 000    |        |        |  |
| 3660 | 015363 | 052    | 043440 | 050101 | A.T016: .ASCIZ '* GAP SIZE-STOP HALF'<HT>    |
| 3661 | 015370 | 051440 | 055111 | 026505 |  |
| 3662 | 015376 | 052123 | 050117 | 044040 |  |
| 3663 | 015404 | 046101 | 004506 | 000    |  |
| 3664 | 015411 | 052    | 043440 | 050101 | A.T017: .ASCIZ '* GAP SIZE-START HALF'<HT>   |
| 3665 | 015416 | 051440 | 055111 | 026505 |  |

E07

DZTEE-A TMO3/TE16 DRIVE FUNCTION TIMER  
DZTEEA.P11 31-MAR-77 00:00

MACY11 30(1046) 27-JUN-77 17:03 PAGE 81  
PROGRAM MESSAGES

|      |        |        |        |        |  |
|------|--------|--------|--------|--------|--|
| 3666 | 015424 | 052123 | 051101 | 020124 |  |
| 3667 | 015432 | 040510 | 043114 | 000011 |  |
| 3668 | 015440 | 020052 | 040507 | 020120 | A.T020: .ASCIZ '* GAP SIZE-INTERRECORD' <HT> |
| 3669 | 015446 | 044523 | 042532 | 044455 |  |
| 3670 | 015454 | 052116 | 051105 | 042522 |  |
| 3671 | 015462 | 047503 | 042122 | 000011 |  |
| 3672 | 015470 | 020052 | 040507 | 020120 | A.T021: .ASCIZ '* GAP CONSISTANCY' <HT>      |
| 3673 | 015476 | 047503 | 051516 | 051511 |  |
| 3674 | 015504 | 040524 | 041516 | 004531 |  |
| 3675 | 015512 | 000    |        |        |  |
| 3676 | 015513 | 052    | 042040 | 052101 | A.T022: .ASCIZ '* DATA TIME-800BPI' <HT>     |
| 3677 | 015520 | 020101 | 044524 | 042515 |  |
| 3678 | 015526 | 034055 | 030060 | 050102 |  |
| 3679 | 015534 | 004511 | 000    |        |  |
| 3680 | 015537 | 052    | 042040 | 052101 | A.T023: .ASCIZ '* DATA TIME-1600BPI' <HT>    |
| 3681 | 015544 | 020101 | 044524 | 042515 |  |
| 3682 | 015552 | 030455 | 030066 | 041060 |  |
| 3683 | 015560 | 044520 | 000011 |        |  |
| 3684 | 015564 | 020052 | 051105 | 051501 | A.T024: .ASCIZ '* ERASE GAP TIME' <HT>       |
| 3685 | 015572 | 020105 | 040507 | 020120 |  |
| 3686 | 015600 | 044524 | 042515 | 000011 |  |
| 3687 | 015606 | 020052 | 051127 | 052111 | A.T025: .ASCIZ '* WRITE FILE MARK' <HT>      |
| 3688 | 015614 | 020105 | 044506 | 042514 |  |
| 3689 | 015622 | 046440 | 051101 | 004513 |  |
| 3690 | 015630 | 000    |        |        |  |
| 3691 | 015631 | 052    | 052040 | 050101 | A.T026: .ASCIZ '* TAPE SPEED-FWD' <HT>       |
| 3692 | 015636 | 020105 | 050123 | 042505 |  |
| 3693 | 015644 | 026504 | 053506 | 004504 |  |
| 3694 | 015652 | 000    |        |        |  |
| 3695 | 015653 | 052    | 052040 | 050101 | A.T027: .ASCIZ '* TAPE SPEED-REV' <HT>       |
| 3696 | 015660 | 020105 | 050123 | 042505 |  |
| 3697 | 015666 | 026504 | 042522 | 004526 |  |
| 3698 | 015674 | 000    |        |        |  |
| 3699 |        |        |        |        |  |
| 3700 | 015675 | 015    | 057012 | 000107 | L.CNTG: .ASCIZ <CR><LF>'↑G'                  |
| 3701 | 015702 | 005015 | 053523 | 036522 | L.SWR: .ASCIZ <CR><LF>'SWR='                 |
| 3702 | 015710 | 000    |        |        |  |
| 3703 | 015711 | 040    | 047040 | 053505 | L.NEW: .ASCIZ ' NEW= '                       |
| 3704 | 015716 | 020075 | 000    |        |  |
| 3705 | 015721 | 015    | 037412 | 005015 | L.QUEST: .ASCIZ <CR><LF>'?' <CR><LF>         |
| 3706 | 015726 | 000    |        |        |  |
| 3707 |        | 015730 |        |        | .EVEN  |
| 3708 |        | 015730 |        |        | ROBUF=.                                      |
| 3709 |        | 015730 |        |        | WTBUF=.                                      |
| 3710 | 015730 | 000200 |        |        | .BLKW 128.                                   |
| 3711 |        | 000001 |        |        | .END   |

|         |          |         |          |         |          |         |          |         |           |
|---------|----------|---------|----------|---------|----------|---------|----------|---------|-----------|
| A       | 010714   | CNTRLU= | 000025   | E.HDR2  | 014320   | L.HDR3  | 014660   | READ    | 005332    |
| ACCL    | = 100000 | CNVDEC  | 002526   | E.NAVA  | 014114   | L.NEW   | 015711   | RESVEC= | 000010    |
| ANGTAB  | 001412   | CNVOCF  | 002420   | E.NCON  | 014015   | L.QUES  | 015721   | REVRD   | 005350    |
| AS      | = 000016 | CNVTAO  | 003056   | E.NDRV  | 014062   | L.RNG   | 014745   | RHINIT  | 005070    |
| ASFLG   | 001130   | CNVTD   | 002540   | E.NSLV  | 014076   | L.SLV   | 014636   | RMR     | = 000004  |
| ATA     | = 100000 | CNVTO   | 002432   | E.SFT   | 014211   | L.SWR   | 015702   | RSTRT   | 006726    |
| ATIME   | 001012   | CR      | = 000015 | E.TIME  | 014373   | MCPE    | = 020000 | RWD     | = 000006  |
| ATIMTB  | 001014   | CRLF    | 001374   | E.TIMO  | 014346   | MDPE    | = 000400 | RWDOFF= | 000002    |
| A.T001  | 014765   | CSITH   | = 002000 | E.TRP4  | 013776   | MMVEC   | = 000250 | R10     | = %000000 |
| A.T002  | 015007   | CS1     | = 000000 | E.UNIT  | 014147   | MOL     | = 010000 | R11     | = %000001 |
| A.T003  | 015027   | CS2     | = 000010 | FC      | = 000006 | MR      | = 000024 | R12     | = %000002 |
| A.T004  | 015051   | DASH    | 001405   | FCE     | = 001000 | MXF     | = 001000 | R13     | = %000003 |
| A.T005  | 015075   | DB      | = 000022 | FINISH  | 012466   | M.EOP   | 013406   | R14     | = %000004 |
| A.T006  | 015117   | DCONST  | 002634   | FMT     | = 000020 | M.EOT   | 013756   | R15     | = %000005 |
| A.T007  | 015136   | DELAY   | 004634   | FPEVEC= | 000244   | M.HSWR  | 013425   | SC      | = 100000  |
| A.T010  | 015160   | DELAYV  | 004664   | FRMCNT= | 177400   | M.NAM   | 013252   | SCOPE   | = 104000  |
| A.T011  | 015203   | DELTIM  | 001114   | FWDSPC  | 005366   | NAMPTR  | 001656   | SCPADR  | 001002    |
| A.T012  | 015225   | DIGTAB  | 001132   | GAP     | 001120   | NED     | = 010000 | SDMN    | = 000020  |
| A.T013  | 015252   | DIVIDE  | 004722   | GAPOK   | 004506   | NEF     | = 004000 | SKEWTS  | 012706    |
| A.T014  | 015301   | DLT     | = 100000 | GAPTBL  | 001054   | NEM     | = 004000 | SLA     | = 000001  |
| A.T015  | 015332   | DPR     | = 000400 | GO      | = 000001 | NOP     | = 000000 | SLAVES  | 006350    |
| A.T016  | 015363   | DRIVES  | 006104   | GTIMTB  | 001556   | NORM11= | 001700   | SLR     | = 177774  |
| A.T017  | 015411   | DRVAVA  | 005012   | GTSWR   | 002020   | NSG     | = 000400 | SLVAVA  | 005040    |
| A.T020  | 015440   | DRVNUM  | 001004   | HERE    | 012652   | OCTALO  | 001116   | SLVNUM  | 001005    |
| A.T021  | 015470   | DRVTL   | 001154   | HLT     | = 104400 | ODIGIT  | 001144   | SLVPTR  | 001006    |
| A.T022  | 015513   | DRY     | = 000200 | HT      | = 000011 | OPI     | = 020000 | SLVTBL  | 001164    |
| A.T023  | 015537   | DRYCLR= | 000010   | IDB     | = 000010 | OR      | = 000200 | SN      | = 000030  |
| A.T024  | 015564   | DS      | = 000012 | IE      | = 000100 | OSC     | = 000100 | SNPT    | 005470    |
| A.T025  | 015606   | DT      | = 000026 | ILF     | = 000001 | OUTBUF= | 005620   | SPACE   | 001410    |
| A.T026  | 015631   | DTE     | = 010000 | ILR     | = 000002 | OUTGAP  | 002744   | SPACE2  | 001407    |
| A.T027  | 015653   | DVA     | = 004000 | INBUF   | 001264   | OUTSPC  | 002650   | SPCFWD= | 000030    |
| A16     | = 000400 | DVD     | = 000000 | INCVAE= | 000100   | PARVEC= | 000114   | SPCREV= | 000032    |
| A17     | = 001000 | DV1     | = 000001 | INIT    | 005620   | PAT     | = 000020 | SPR     | = 002000  |
| BA      | = 000004 | DV2     | = 000002 | IOTVEC= | 000020   | PEFLRC= | 000200   | SSC     | = 000100  |
| BAI     | = 000010 | DV3     | = 000003 | IR      | = 000100 | PES     | = 000040 | STIMTB  | 001416    |
| BEGIN   | 006750   | DV4     | = 000004 | ITCNT   | 001121   | PE1600= | 002300   | STKPTR= | 000600    |
| BELL    | 001403   | DV5     | = 000005 | I.DRV   | 013656   | PFVEC   | = 000024 | SWR     | 001000    |
| BKSLSH  | 001377   | DV6     | = 000006 | I.DRVS  | 013567   | PGE     | = 002000 | SWREG   | 000176    |
| BOT     | = 000002 | DV7     | = 000007 | I.REG   | 013522   | PIP     | = 020000 | SW06    | = 000100  |
| BPI200= | 000000   | ECHO    | 001401   | I.REM   | 013455   | PIRQ    | = 177772 | SW07    | = 000200  |
| BPI556= | 000400   | EMTVEC= | 000030   | I.SLVS  | 013637   | PIRVEC= | 000240   | SW08    | = 000400  |
| BPI800= | 001000   | END     | 012570   | I.SPD   | 013723   | PLKCSR= | 172540   | SW09    | = 001000  |
| BPTVEC= | 000014   | EOT     | = 002000 | LF      | = 000012 | PLKVEC= | 000104   | SW10    | = 002000  |
| CDM11   | = 000320 | ER      | = 000014 | LKS     | = 177546 | PRGFLG  | 001124   | SW11    | = 004000  |
| CHKDRV  | 006250   | ERASE   | = 000024 | LKVEC   | = 000100 | PSCNT   | 001127   | SW13    | = 020000  |
| CHKSLV  | 006560   | ERFLG   | 001123   | LPB     | = 177516 | PSEL    | = 002000 | SW14    | = 040000  |
| CHNFLG  | 005716   | ERR     | = 040000 | LPS     | = 177514 | PSW     | = 177776 | SW15    | = 100000  |
| CH7     | = 010000 | ERRTRP  | 003542   | L.ACT   | 014755   | PUBLIS  | 003144   | TAP     | = 040000  |
| CLR     | = 000040 | ERRVEC= | 000004   | L.CHAN  | 014641   | RDBUF   | = 015730 | TBITVE= | 000014    |
| CNTRLA= | 000001   | E.DRV   | 014104   | L.CNTG  | 015675   | RDFWD   | = 000070 | TC      | = 000032  |
| CNTRLC= | 000003   | E.GAP   | 014434   | L.DRV   | 014624   | RDREV   | = 000076 | TCRLF   | 002156    |
| CNTRLG= | 000007   | E.HDR   | 014235   | L.HDR1  | 014444   | RDSW    | 002126   | TIB     | 002016    |
| CNTRLO= | 000017   | E.HDR1  | 014245   | L.HDR2  | 014557   | RDY     | = 000200 | TIMER   | 004340    |



|        |        |         |        |        |        |         |         |        |         |        |        |
|--------|--------|---------|--------|--------|--------|---------|---------|--------|---------|--------|--------|
| TIMERR | 004350 | TRTVEC= | 000014 | TST017 | 011224 | UNS     | =       | 040000 | \$CRLF  | 002122 |        |
| TIMERO | 004364 | TSTNUM  | 001122 | TST020 | 011334 | UNTFND  | =       | 001125 | \$ENDAD | 012642 |        |
| TIMERI | 004314 | TSTTBL  | 001736 | TST021 | 011454 | UPE     | =       | 020000 | \$FILL  | 002111 |        |
| TIMOK  | 004376 | TST000  | 007166 | TST022 | 011760 | WAITRO  | =       | 005126 | \$HT    | =      | 000011 |
| TIMON  | 004250 | TST001  | 007432 | TST023 | 012110 | WAITTI  | =       | 005130 | \$NULL  | =      | 002110 |
| TKB    | =      | TST002  | 007516 | TST024 | 012240 | WC      | =       | 000002 | \$SVPC  | =      | 000040 |
| TKB    | =      | TST003  | 007574 | TST025 | 012360 | WCE     | =       | 040000 | \$TKFLG | =      | 002113 |
| TKISR  | 003420 | TST004  | 007664 | TST026 | 012714 | WCHKF   | =       | 000050 | \$TPB   | =      | 002116 |
| TKS    | =      | TST005  | 007772 | TST027 | 013042 | WCHKR   | =       | 000056 | \$TPFLG | =      | 002112 |
| TKVEC  | =      | TST006  | 010056 | TYPDEC | 002534 | WFMK    | =       | 000026 | \$TPS   | =      | 002114 |
| TMBASE | 001010 | TST007  | 010142 | TYPE   | =      | 000060  | WFWD    | =      | 016330  | =      | 016330 |
| TMCMD  | 005450 | TST010  | 010242 | TYPE1  | 002150 | WRDCNT  | =       | 177600 | .HLT    | =      | 003544 |
| TMCSI  | =      | TST011  | 010362 | TYPE2  | 002176 | WRITE   | =       | 005314 | .INPUT  | =      | 003272 |
| TMK    | =      | TST012  | 010460 | TYPE3  | 002204 | WRL     | =       | 004000 | .RESTO  | =      | 002376 |
| TPB    | =      | TST013  | 010570 | TYPE4  | 002210 | WRT.BK  | =       | 005406 | .REWIND | =      | 005232 |
| TPS    | =      | TST014  | 010730 | TYPFLG | 001126 | WTBUF   | =       | 015730 | .SAVE   | =      | 002354 |
| TPVEC  | =      | TST015  | 011022 | TYPOCT | 002426 | \$CHARC | =       | 002120 | .SCOPE  | =      | 004010 |
| TRAPVE | =      | TST016  | 011130 | UBREAK | =      | 002121  | \$CNTRL | =      | .TYPE   | =      | 002130 |
| TRE    | =      |         |        |        |        |         |         |        |         |        |        |

. ABS. 016330 000

ERRORS DETECTED: 0

DZTEEA, DZTEEA/SOL+DZTEEA.SML/ML, DZTEEA.P11  
RUN-TIME: 4 6 .3 SECONDS  
RUN-TIME RATIO: 147/11=13.1  
CORE USED: 7K (13 PAGES)