IDENTIFICATION

PRODUCT CODE: AC-9431D-MC
PRODUCT NAME: CZTRBD0 TR79 UTILITY PROGRAMS
PRODUCT DATE: FEBURYARY 1982
MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND
MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR
RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976, 1982
DIGITAL EQUIPMENT CORPORATION, MAYNARD MASS.
TR79 UTILITY PROGRAM

TABLE OF CONTENTS

SECTION                     SUBJECT
1.                          ABSTRACT
2.                          REQUIREMENTS
3.                          LOADING PROCEDURE
4.                          STARTING PROCEDURE
5.                          RESTRICTIONS
6.                          CONSOLE SETTINGS
7.                          OPERATION
8.                          LISTING

1. ABSTRACT

This program is in two parts, and is intended to provide the user with a tool for trouble-shooting the TR79 MAGTAPE subsystem on a PDP-11 computer system. The first part of the program allows the user to give the MAGTAPE commands, to simulate user routines by merely inserting these commands in the core locations provided. The user may execute one or several instructions in any legal sequence, while the code for the driver is simple and uses no interrupts, due to the design of the hardware. Certain error conditions must be identified in order to prevent misinterpretation of the desired results.

Part two of the program consists of self contained routines to permit the user to set up and check the delays contained within the TR79 controller, by using the switch register to select the appropriate routine.
2. REQUIREMENTS

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

2.2 STORAGE
   THIS PROGRAM REQUIRES A MINIMUM OF 4K OF CORE

---------------------------------------------------------------

3. LOADING
   USE STANDARD BINARY LOADING PROCEDURE

---------------------------------------------------------------

4.0 STARTING PROCEDURE
   THERE ARE TWO STARTING Addresses THAT MAY BE USED

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

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

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

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

1. ATTEMPT TO WRITE DATA FROM LOAD POINT WITHOUT AN I.D.B.
2. ATTEMPT TO WRITE A TAPE MARK FROM LOAD POINT
3. ATTEMPT TO MOVE TAPE IN REVERSE FROM LOAD POINT
4. ATTEMPT TO WRITE AN I.D.B. AT OTHER THAN LOAD POINT
5. ATTEMPT TO WRITE DATA WITH THE WRITE RING REMOVED
6. COMMAND WHILE THE MAGTAPE IS NOT READY
7. ISSUE A COMMAND WHILE THE CONTROLLER IS OUT OF ORDER
8. ISSUE A COMMAND WITH INHIBIT BIT SET
9. ILLEGAL FUNCTION CODES 00, 03, 05, 06, 11, 12, 14

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

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

SW 15 = 1 STOP AFTER EACH OPERATION (ONLY WITH START 200)
0 PROCEED

SW 14 = 1 STOP AT THE END OF EACH PROGRAM PASS (ONLY WITH START 200)
0 PROCEED

SW 7 = 1 ENABLE FOR DELAY Routines (EXECUTE ROUTINE ONLY WITH START 204)
0 ALLOW SELECTION OF DELAY Routines WITH SW 0-3

SW 0 THRU 3 = DELAY ROUTINE TO BE EXECUTED (ONLY WITH START 204)

6.1 DELAY SETUP TABLE

<table>
<thead>
<tr>
<th>SWITCH SETTING</th>
<th>DELAY NAME</th>
<th>MOD/C</th>
<th>LOCATION</th>
<th>PAGE</th>
<th>PIN</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>NO-OP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>P CLR</td>
<td>M-302</td>
<td>C-06</td>
<td>T02-2</td>
<td>H2</td>
<td>F2</td>
</tr>
<tr>
<td>02</td>
<td>P CLR OFF</td>
<td>M-302</td>
<td>D-09</td>
<td>T04-1</td>
<td>H2</td>
<td>T2</td>
</tr>
<tr>
<td>03</td>
<td>ERROR CLK</td>
<td>M-302</td>
<td>C-10</td>
<td>T04-2</td>
<td>H2</td>
<td>F2</td>
</tr>
<tr>
<td>04</td>
<td>WRITE ENAB</td>
<td>M-302</td>
<td>C-10</td>
<td>T09-3</td>
<td>M2</td>
<td>T2</td>
</tr>
<tr>
<td>05</td>
<td>BUFF CONT</td>
<td>M-302</td>
<td>B-18</td>
<td>T11-1</td>
<td>E1</td>
<td>J1</td>
</tr>
<tr>
<td>06</td>
<td>END WR DAT</td>
<td>M-302</td>
<td>A-16</td>
<td>T11-1</td>
<td>M2</td>
<td>T2</td>
</tr>
<tr>
<td>07</td>
<td>1ST WD REQ</td>
<td>M-302</td>
<td>A-22</td>
<td>T11-2</td>
<td>H2</td>
<td>F2</td>
</tr>
<tr>
<td>09</td>
<td>ERASE</td>
<td>M-302</td>
<td>A-25</td>
<td>T09-3</td>
<td>S1</td>
<td>M1</td>
</tr>
<tr>
<td>10</td>
<td>WRITE IDB</td>
<td>M-302</td>
<td>A-16</td>
<td>T09-1</td>
<td>H2</td>
<td>F2</td>
</tr>
<tr>
<td>12</td>
<td>IDB TIMING</td>
<td>M-302</td>
<td>D-13</td>
<td>T09-1</td>
<td>H2</td>
<td>F2</td>
</tr>
<tr>
<td>13</td>
<td>ABORT</td>
<td>M-302</td>
<td>A-16</td>
<td>T09-3</td>
<td>H2</td>
<td>T2</td>
</tr>
<tr>
<td>14</td>
<td>BUSY DELAY</td>
<td>M-304</td>
<td>B-18</td>
<td>T05-1</td>
<td>R1</td>
<td>P1</td>
</tr>
<tr>
<td>15</td>
<td>GO BIT DEL</td>
<td>M-304</td>
<td>B-18</td>
<td>T06-1</td>
<td>D1</td>
<td>H1</td>
</tr>
<tr>
<td>16</td>
<td>M.S.D.</td>
<td>M-302</td>
<td>A-22</td>
<td>T09-2</td>
<td>M2</td>
<td>T2</td>
</tr>
<tr>
<td>17</td>
<td>NO-OP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.2 DELAY

$\$ 00 NO OPERATION PERFORMED WAITING SWITCH SELECTION AND ENABLE

$\$ 01 POWER CLEAR PERFORMED NO TAPE MOTION DELAY PULSE IS POSITIVE GOING

$\$ 02 POWER CLEAR PERFORMED NO TAPE MOTION DELAY PULSE IS POSITIVE GOING

$\$ 03 NO TAPE MOTION, PROGRAM FORCES AN ERROR WITH THE BGL BIT IN THE TR STATUS REGISTER (BIT 11) DELAY PULSE IS POSITIVE GOING

$\$ 04 TAPE MOTION, PROGRAM DOES A SHORT ERASE WHILE MOVING TAPE
TAPE MOTION IS NOT READILY NOTICIBLE WHILE EXECUTING THIS ROUTINE
DELAY PULSE IS POSITIVE GOING

$\$ 05 TAPE MOTION, PROGRAM DOES A 10 BYTE WRITE. PROGRAM CHECKS
FOR LOAD POINT AND WILL WRITE AN I.D.B. BEFORE ENTERING THE
DELAY LOOP. DELAY PULSE IS POSITIVE GOING

$\$ 06 SAME CONDITIONS AS 05 DELAY PULSE IS POSITIVE GOING

$\$ 07 SAME CONDITIONS AS 05 DELAY PULSE IS POSITIVE GOING

$\$ 10 TAPE MOTION, PROGRAM WILL CHECK FOR LOAD POINT THEN DO A
MAXIMUM ERASE TO MAKE THE OPERATION CONTINUOUS THE PROGRAM
WILL CLEAR THE ERASE COUNT BEFORE THE OPERATION IS DONE.
DELAY PULSE IS NEGATIVE GOING

$\$ 11 TAPE MOTION, PROGRAM WILL CONTINUOUSLY WRITE THE I.D.B.
DELAY PULSE IS POSITIVE GOING

$\$ 12 SAME CONDITIONS AS 11. DELAY PULSE IS POSITIVE GOING

$\$ 13 PROGRAM WILL REWIND TAPE TO L.P. AND FORCE AN ERROR BY DOING
A WRITE DATA. DELAY PULSE IS POSITIVE GOING.

$\$ 14 TAPE WILL MOVE TO L.P., AND DO A MAXIMUM ERASE.
WHILE THIS IS HAPPENING PROGRAM WILL LOAD THE COMMAND
REGISTER TO PRODUCE A LD CTRL PULSE. DELAY PULSE IS POSITIVE GOING

$\$ 15 SAME CONDITIONS AS 04. DELAY PULSE IS NEGATIVE GOING

$\$ 16 PROGRAM WILL MOVE TAPE TO E.O.T. AND ATTEMPT TO DO A FAST
FORWARD TO PRODUCE THE MOTION STOP DELAY. DELAY IS POSITIVE.
NOTE: AFTER COMPLETION OF THIS ROUTINE A MANUAL REWIND
SHOULD BE PERFORMED.

17 THIS IS A NO OPERATION SAME AS 00
7.0 OPERATION

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

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

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Core Address</td>
<td>700</td>
<td>Bits 12 and 13 of this location represent XBA 16 and XBA 17 of the TA control register. These bits allow relocation of the data buffer.</td>
</tr>
<tr>
<td>Unit Select</td>
<td>702</td>
<td>Bits 8-9 in this location represent the unit numbers of the tape drives. A maximum of 4 drives per controller default is unit 0.</td>
</tr>
<tr>
<td>Word Count</td>
<td>704</td>
<td>This is the 2's complement of the number of words transferred. Since each byte occupies a word location the number of core locations used is 2X the word count. Program default is -20 words.</td>
</tr>
<tr>
<td>Read Address</td>
<td>706</td>
<td>Contains address of the read buffer. Program default is location 6700.</td>
</tr>
<tr>
<td>Write Address</td>
<td>710</td>
<td>Contains address of the write buffer. The program contains 4 write patterns contiguous in core. Location 2700 = all 1's pattern Location 3700 = alternate 1 and 0 bytes Location 4700 = alternate 1 and 0 bits Location 5700 = sliding 1 bit pattern Program default is location 2700</td>
</tr>
<tr>
<td>Erase Count</td>
<td>712</td>
<td>Contains a 2's complement number proportional to the amount of tape to be erased. This number is loaded into the word count register prior to an erase command being performed. Program default is 7777. Each increment causes .02 inches of tape to be erased.</td>
</tr>
<tr>
<td>Operation Delay</td>
<td>714</td>
<td>Contains a number used in a timer between operations default =000001 minimum delay</td>
</tr>
<tr>
<td>Operation Delay Mult.</td>
<td>716</td>
<td>This is used in conjunction with loc. 714 as a multiplier in the delay timer. Default is 000004 minimum delay. Increasing this number will allow more time between operations.</td>
</tr>
<tr>
<td>Operations Number</td>
<td>720</td>
<td>This location contains the number of operations to be performed in the op table. Default = 1.</td>
</tr>
<tr>
<td>Operations Table</td>
<td>722</td>
<td>This is the beginning of the operations table. All operations to be performed should be entered in the desired sequence in this table. Default is a write operation.</td>
</tr>
<tr>
<td>Thru</td>
<td>766</td>
<td></td>
</tr>
</tbody>
</table>
### 7.2 Error Checks and Halts

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1320</td>
<td><strong>Halt Here if there was an attempt to execute an illegal function.</strong> Due to tape position or sequence of instructions, the illegal command is displayed in RO when the program halts. See Section 3.1B for illegal functions</td>
</tr>
<tr>
<td>1332</td>
<td><strong>Halt here if there was a hardware error on the previous operation if it is desired to bypass the error flag</strong> at this location. The command and status register should be examined at this time to determine the probable cause of the error. Pressing continue will clear the error by executing a control reset.</td>
</tr>
<tr>
<td>1350</td>
<td><strong>Halt here if your operation table loc. 722-766 has an operation that is not defined in the legal function codes.</strong> RO has the bad code in it. Check your table in locations 722 thru 766.</td>
</tr>
<tr>
<td>1406</td>
<td><strong>Halt here if bit 15 of the SWR is set. This is the halt between instructions.</strong></td>
</tr>
<tr>
<td>1432</td>
<td><strong>Halt here if bit 14 of the SWR is set. This is the halt between passes of instructions in the OP table.</strong></td>
</tr>
</tbody>
</table>
7.3 TABLE OF LEGAL FUNCTIONS AND CODES FOR USE IN OPERATIONS TABLE (LOC 722-766)

<table>
<thead>
<tr>
<th>CODE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td><strong>ILLEGAL</strong></td>
</tr>
<tr>
<td>01</td>
<td>WRITE DATA (ILLEGAL IF EXECUTED FROM LOAD POINT)</td>
</tr>
<tr>
<td>02</td>
<td>READ (DATA, TAPE MARK OR I.D.B.)</td>
</tr>
<tr>
<td>03</td>
<td><strong>ILLEGAL</strong></td>
</tr>
<tr>
<td>04</td>
<td>SPACE REVERSE (ILLEGAL IF ISSUED FROM LOAD POINT)</td>
</tr>
<tr>
<td>05</td>
<td><strong>ILLEGAL</strong></td>
</tr>
<tr>
<td>06</td>
<td><strong>ILLEGAL</strong></td>
</tr>
<tr>
<td>07</td>
<td>ERASE</td>
</tr>
<tr>
<td>10</td>
<td>REWIND (TAPE MOVES AT 160 I.P.S.) (ILLEGAL IF ISSUED FROM LOAD POINT.</td>
</tr>
<tr>
<td>11</td>
<td><strong>ILLEGAL</strong></td>
</tr>
<tr>
<td>12</td>
<td><strong>ILLEGAL</strong></td>
</tr>
<tr>
<td>13</td>
<td>FAST FORWARD (TAPE MOVES FORWARD AT 160 I.P.S.)</td>
</tr>
<tr>
<td>14</td>
<td><strong>ILLEGAL</strong></td>
</tr>
<tr>
<td>15</td>
<td>WRITE I.D.B. (ILLEGAL IF ISSUED AT OTHER THAN LOAD POINT)</td>
</tr>
<tr>
<td>16</td>
<td>WRITE TAPE MARK (ILLEGAL IF ISSUED FROM LOAD POINT)</td>
</tr>
<tr>
<td>17</td>
<td>OFFLINE (REQUIRES MANUAL INTERVENTION)</td>
</tr>
<tr>
<td>20</td>
<td>CONTROL RESET (PROGRAM PSEUDO OP)</td>
</tr>
</tbody>
</table>
8. PROGRAM LISTING

.ENABLE ABS,AMA
.TITLE TR79 UTILITY DRIVER
.ASECT

***********************************************************************
* GENERAL REGISTER DEFINITIONS *
***********************************************************************

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

***********************************************************************
* TR79 REGISTER DEFINITIONS *
***********************************************************************

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

***********************************************************************
* PROCESSOR REGISTER DEFINITIONS *
***********************************************************************

PSW=177776
SWR=177570

***********************************************************************
* TTY REGISTERS *
***********************************************************************

TTSTAT=177564
TTBUF=177566
***************
* TRAP CATCHERS *
***************

; 000000 =0
'REPT 200
.+2
HALT
.ENDED

***************
* STARTS AND CORE BUFFER BUILD *
***************

; 00200 
BEGIN:
JMP @CORBIL ;NORMAL START
INC @NWORST ;SETS FLAG TO DETERMINE WHO STARTED
JMP @CORBIL ;START HERE FOR DELAY ROUTINES

00200 000137 000230 BEGIN:
00214 000777
00216 000777
00220 000400
00222 000777
00224 000525
00226 000652
00230 012701 000214 CORBIL:
00234 012703 000216

00240 012702 002700 X1:
00244 012700 177400 X2:
00250 011122 X3:
00252 005200
00254 001403
00256 011322
00260 005200
00262 001372
00264 062701 000004 X4:
00270 062703 000004
00274 022701 000230

MOV #214,R1 ;SET UP PATTERN GENERATOR
MOV #216,R3 ;SET UP PATTERN GENERATOR
MOV #2700,R2 ;SET UP ADDRESS POINTER
MOV #177400,R0 ;SET UP COUNTER
MOV (R1),(R2)+ ;DO IT
INC R0 ;KEEP TRACK OF HOW MANY
B EQ X4 ;CHECK FOR DONE
MOV (R3),(R2)+ ;DO IT
INC R0 ;KEEP COUNTING
B NE X3 ;LOOP HERE
ADD #4,R1 ;NEXT PATTERN
ADD #4,R3 ;GO TO NEXT PATTERN
CMP #230,R1 ;DONE YET
*************** PROGRAM CONSTANTS AND VARIABLES ***************

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

*************** TABLE OF OPERATIONS TO BE PERFORMED MAX =23 ***************

<table>
<thead>
<tr>
<th>OPTBL</th>
<th>1</th>
<th>1ST OPERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2ND OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3RD OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>6TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>8TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>9TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>10TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>11TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>12TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>13TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>14TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>15TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>16TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>17TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>18TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>19TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>20TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>21ST OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>22ND OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>23RD OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>24TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>25TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>26TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>27TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>28TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>29TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>30TH OPERATION</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>31ST OPERATION</td>
<td></td>
</tr>
</tbody>
</table>

***************
**HOUSE KEEPING AND INITIAL PROGRAM PARAMETERS**

---

START: MOV #600, SP ; SET UP STACK AT LOC. 600
MOV #360, @PSW ; SET PRIORITY LEVEL 7 NO INTERRUPTS ALLOWED
MOV #4000, @TRCOM ; WAIT FOR THE INITIAL 900 MS. POWER CLEAR TO SUBLIDE
BIT 1$ ; BNE 1$ ; LOOP UNTILL CONTROL IS READY
MOV @OPNUM, P1 ; R1 CONTAINS HOW MANY OPERATIONS WILL BE DONE
MOV #76377, @UNIT ; STRIP OFF EVERYTHING EXCEPT BITS 8 AND 9 TO SET UP UNI

---

**SET UP OPERATIONS, COMPARATOR**

---

MOV #OPTBL, R2 ; R2 CONTAINS THE POINTER TO THE START OF THE OPERATIONS
MOV #9, @TEMP1 ; THE NUMBER OF LEGAL OPERATIONS
MOV #LEGOPS, R3 ; POINTER TO THE BEGINNING OF THE LEGAL OPERATIONS COMPA
MOV (R2)+, @TEMP2 ; GET THE OPERATION AND PUT IT IN TEMP 2
CMR @TEMP2, (R3)+ ; CHECK IT AGAINST THE LEGAL FUNCTIONS
BEQ CLINP2 ; IF ITS LEGAL CONTINUE TO EXECUTE IT
DEC TEMP1 ; WASN'T THAT OPERATION SUBTRACT 1
BNE 2$ ; TRY NEXT ONE
**CHECK FOR A CONTROL RESET**

* MUST BE A CONTROL RESET JUMP TO CRES
* IT MUST BE ILLEGAL FUNCTION STOP THE PROGRAM
* MUST BE A POWER CLEAR SO DO IT TAKES 900 MILI SECONDS
* SEE IF DONE WITH POWER CLEAR YET
* WAIT UNTILL DONE
* GET BACK INTO PROGRAM

**CLEAR INHIBIT BIT AND SET UP OPERATION**

* CLEAR OUT THE INHIBIT BIT
* GET AN OFFSET VALUE
* ADD IT TO THE PC AND GO THERE

**OPERATIONS DIRECTORY TABLE**

* WRITE INSTRUCTION
* READ INSTRUCTION
* SPACE REVERSE INSTRUCTION
* ERASE COMMAND
* REWIND COMMAND
* FAST FORWARD COMMAND
* WRITE ID BURST
* WRITE TAPE MARK
* SET UP UNIT NUMBER AND GO BIT *

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

* OPERATIONS SETUP ROUTINES *

WRITE: MOV @HADDR, @TRBA ; SET UP WRITE BUFFER AREA
BR WR1WC ; CONTINUE ON TO SET UP W.C.
READ: MOV @HADDR, @TRBA ; SET UP READ BUFFER ADDRESS
WR1WC: MOV @WCNT, @TRWC ; SET UP WORD COUNT
BR SGOB ; SET UP GO BIT
ERSE: MCV @ERSCNT, @TRWC ; SET NUMBER TO INDICATE AMOUNT OF TAPE TO BE ERASED
BR SGOB ; SET UP GO BIT

* ROUTINE TO EXECUTE THE FUNCTION AND CHECK FOR DONE *

EXECUT: NOP
MOV @TEMP2, @TRCOM ; DO THE FUNCTION
7$: TSTB @TRCOM ; SEE IF DONE YET
BPL 7$ ; WAIT FOR IT
* ROUTINE TO CHECK FUNCTION WHEN DONE *

FUDONE:  MOV @TRSTAT,@STITEM  ;SAVE STATUS
         MOV @TRCOM,@COMTEM  ;SAVE COMMAND REGISTER
         BIT #40000,@COMTEM  ;WAS IT AN ILLEGAL COMMAND DUE TO SEQUENCE OR TAPE POSI
         BEQ ERDONE  ;NO ERROR HERE
         ILLCOM:  ASR @TEMP2  ;STRIP OFF THE GO BIT
                   MOV @TEMP2,RO  ;PUT BAD COMMAND IN RO
                   ERR14:  HALT  ;STOP WITH BAD COMMAND DISPLAYED
                   ERDONE:  BIT #100000,@COMTEM  ;SEE IF ERROR BIT IS SET
                                 BEQ OPDEL  ;NO ERRORS CONTINUE
                   ERR15:  HALT  ;GOT AN ERROR NOP THIS HALT TO CONTINUE
                   JMP @RES  ;IF YOU GOT AN ERROR ONLY RECOVERY IS WITH A CONTROL RE
                   ILLFUN:  ASR @TEMP2  ;STRIP OFF GO BIT
                             ILLFUT:  MOV @TEMP2,RO  ;PUT THE BAD CODE IN RO TO DISPLAY WHEN Halted
                             ERRIF:  HALT  ;GOT AN ILLEGAL FUNCTION CHECK YOUR PROGRAM LOCATION 72

783 784 785 786 787 788 789 001352 013737 000716 000666 OPDEL: MOV @#OPDLX,#TIMMUL :SET OP OPERATIONS DELAY MULTIPLIER
790 001360 013700 000714 MOV @#OPDLY,RO :SET UP OPERATIONS DELAY TIMER
791 001364 005300 BNE 8$ :TIMER IS TICKING
792 001366 001376 DEC RO :GET MORE TIME
793 001370 005337 000666 BNE 8$ :GET MORE TIME
794 001374 001373 DEC @TIMMUL :COUNT DOWN THE MULTIPLIER
795 001376 032737 100000 177570 BNE 8$ :GET MORE TIME
796 001404 002901 100000 177570 TIMES UP CHECK SWITCHES TO SEE IF WE HALT OR CONTINUE
797 001406 000000 BEQ .+4 :DON'T STOP NOW SKIP THE HALT
798 001410 005301 INSHT: HALT :STOP BETWEEN INSTRUCTIONS
799 001412 001001 DEC R1 :-1 FROM THE NUMBER OF OPERATIONS IN R1
800 001414 000401 BNE 9$: :GO AND DO THE NEXT INSTRUCTION
801 001416 000137 001040 BR .+4 :SKIP THE JUMP
802 001422 032737 040000 177570 9$: :DO THE LOOP AGAIN
803 001430 001401 JMP @#LOOP :CHECK SWITCHES TO SEE IF WE WANT TO STOP AT END OF PASS
804 001432 000000 100000 177570 177570 :CHECK SWITCHES TO SEE IF WE WANT TO STOP AT END OF PASS
805 001434 000137 001000 PASHLT: HALT :STOP BETWEEN PASSES
806 001436 000137 001000 REST: JMP @#START :GO DO IT AGAIN (NEXT PASS)

******************************************************
* LEGAL OPERATIONS COMPARATOR TABLE *
******************************************************
810
811
812 001440 000001 LEGOPS: 00001 :WRITE
813 001442 000002 :READ
814 001444 000004 :SPACE REVERSE
815 001446 000007 :ERASE
816 001450 000010 :REWIND
817 001452 000013 :FAST FORWARD
818 001454 000015 :WRITE IDB
819 001456 000016 :WRITE TAPE MARK
820 001460 000017 :OFFLINE
.................................
* DELAY SET-UP PROGRAM ROUTINES *
.................................

322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343

001462
001462
001466
001466
001472
001472
001500
001500
001506
001506
001510
001510
001516
001516
001522
001522
001524
001524
001532
001532
001536
001536
001542
001542

DRTN: MOV #600.SP
:SET STACK
MOV #500.RS
:SET UP SUBROUTINE STACK
MOV #340.@PSW
:SET PRIORITY 7 NO INTERRUPTS ALLOWED
177776
177776
164000
164000
DROUTs: BIT #4000.@TRCOM
:CHECK FOR INITIAL POWER CLEAR TO SUBSIDE
:WAIT TILL DONE
:GET SWITCHES AND PUT THEM IN STORAGE
:SEE IF ENABLE IS UP YET
177570
177570
000664
000664
BRI: MOV @SWR,@SWRTM
TSTB @SWRTM
BPL BRI
:LOOP UNTILL ENABLE IS UP
000664
000664
@SWRTM
@SWRTM
177760
177760
000664
000664
BIC #177760.@SWRTM
ASL @SWRTM
ASL @SWRTM
ADD @SWRTM,PC
:MASK BITS AND GET A NUMBER BETWEEN 0-16
:SHIFT IT LEFT TO MULTIPLY BY 2
:SHIFT AGAIN MULTIPLY BY 2 TO GET OFFSET
:ADD IT TO THE PC AND GO THERE
<table>
<thead>
<tr>
<th>Table</th>
<th>JMP Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>JMP AMBR1</td>
<td>:DELAY 0 IS A NO-OP</td>
</tr>
<tr>
<td>850</td>
<td>JMP AMBR2</td>
<td>:DELAY 1 IS FOR POWER CLEAR</td>
</tr>
<tr>
<td>851</td>
<td>JMP AMBR3</td>
<td>:DELAY 2 IS FOR POWER CLEAR</td>
</tr>
<tr>
<td>852</td>
<td>JMP AMBR4</td>
<td>:DELAY 3 IS FOR ERROR CLK</td>
</tr>
<tr>
<td>853</td>
<td>JMP AMBR5</td>
<td>:DELAY 4 IS FOR WRITE ENABLE</td>
</tr>
<tr>
<td>854</td>
<td>JMP AMBR5</td>
<td>:DELAY 5 IS FOR OUT BUFF FLAG</td>
</tr>
<tr>
<td>855</td>
<td>JMP AMBR5</td>
<td>:DELAY 6 IS FOR END WRITE DATA</td>
</tr>
<tr>
<td>856</td>
<td>JMP AMBR5</td>
<td>:DELAY 7 IS FOR FIRST WORD WRITE REQUEST</td>
</tr>
<tr>
<td>857</td>
<td>JMP AMBR5</td>
<td>:DELAY 10 IS FOR ERASE</td>
</tr>
<tr>
<td>858</td>
<td>JMP AMBR11</td>
<td>:DELAY 11 IS FOR WRITE I.D.B.</td>
</tr>
<tr>
<td>859</td>
<td>JMP AMBR11</td>
<td>:DELAY 12 IS FOR I.D.B. TIMING</td>
</tr>
<tr>
<td>860</td>
<td>JMP AMBR16</td>
<td>:DELAY 13 IS FOR ABORT WINDOW</td>
</tr>
<tr>
<td>861</td>
<td>JMP AMBR6</td>
<td>:DELAY 14 IS FOR LD CTRL + BUSY</td>
</tr>
<tr>
<td>862</td>
<td>JMP AMBR4</td>
<td>:DELAY 15 IS FOR GO PULSE DELAY</td>
</tr>
<tr>
<td>863</td>
<td>JMP AMBR1</td>
<td>:DELAY 16 IS FOR M.S.D. DELAY</td>
</tr>
<tr>
<td>864</td>
<td>JMP AMBR14</td>
<td>:DELAY 17 IS A NO-OP</td>
</tr>
</tbody>
</table>
**ROUTINE FOR DELAYS 1 AND 2**

BR2: JSR R5, @PCL :: DO A POWER CLEAR
    CLR R0 :: CLEAR THE TIMER
    MOV #4, R1 :: SET TIMING DELAY MULTIPLIER
    BR2A: DEC R0 :: TIMER IS TICKING
           BNE BR2A :: WAIT TILL ITS DONE
           MOV #1, R1 :: ONCE THROUGH THE TIMING LOOP
           DEC R1 :: REPEAT LOOP IF MULTIPLIER IS NON ZERO
           BNE BR2A :: CHECK FOR A LOOP
           TSTB @#$WR :: DO IT AGAIN
           JMP @#$BR1 :: GET NEXT TEST

**ROUTINE FOR DELAY 3**

BR3: MOV #70, R0 :: SET UP DELAY MULTIPLIER
    MOV #4, R1 :: SET TIMING DELAY MULTIPLIER
    BIS #4000, @#TRSTAT :: FORCE AN ERROR WITH B.G.L. BIT
    BR3: DEC R0 :: TIMER IS TICKING
           BNE 13$: :: CHECK TIMER
           CLR @#TRSTAT :: OK NOW CLEAR THE BIT
           DEC R1 :: ONCE THROUGH TIMING LOOP
           BNE 13$: :: REPEAT LOOP IF MULTIPLIER IS NON ZERO
           TSTB @#$WR :: SEE IF WE WANT TO DO IT AGAIN
           BMI BR3 :: OK LOOP BACK
           JSR R5, @PCL :: DONE HERE DO A POWER CLEAR AND GET THE NEXT ONE
           JMP @#$BR1 :: GET THE NEXT DELAY DIRECTIVE

**ROUTINE FOR DELAY 4 AND 15**

BR4: JSR R5, @#RDY :: CHECK FOR READY
    BR4A: MOV #1, @#TRWC :: THIS NUMBER IS USED FOR AN ERASE COUNT
           MOV #17, @#TRCOM :: DO THE ERASE
           JSR R5, @#RDY :: WAIT TILL DONE
           JSR R5, @#ERCK :: SEE IF WE ERRORED OUT
           TSTB @#$WR :: SEE IF WE LOOPON TEST
           BMI BR4A :: LOOP HERE AND DO IT AGAIN
           JMP @#$BR1 :: GET OUT AND GET NEXT DELAY DIRECTIVE
915 916 917 918 919
920 002014 004537 002570 BR5: JSR R5, @OFLP ;SEE IF WE ARE AT LOAD POINT
921 002020 005037 164002 CLR @TRSTAT ;CLEAR INHIBIT BIT
922 002024 012737 177774 BR5A: MOV #4, @TRWC ;SET UP FOR A 10 BYTE WRITE
923 002028 013737 000710 164006 MOV @VADDR, @TRBA ;SET UP THE WRITE ADDRESS BUFFER
924 002039 012737 000003 164000 MOV #3, @TRCOM ;DO THE WRITE
925 002046 004537 002522 JSR R5, @RDY ;WAIT FOR READY
926 002052 004537 002540 JSR R5, @ERCK ;CHECK FOR ERRORS
927 002056 105737 177570 TSTB @SWR ;SEE IF WE WANT TO LOOP
928 002062 100760 BMI BR5A ;LOOP HERE AND DO IT AGAIN
929 002064 000137 001510 JMP @#BR1 ;GET OUT AND GET NEXT DELAY DIRECTIVE

930 931 932 933 934
935 936 002070 004537 002624 BR6: JSR R5, @REW ;DO A REWIND GET TO B.O.T.
937 002074 004537 002654 BR6C: JSR R5, @ETS ;CHECK FOR END OF TAPE ANYWAY
938 002100 005037 164002 CLR @TRSTAT ;CLEAR THE INHIBIT BIT
939 002104 005037 164004 CLR @TRWC ;CLEAR THE WORD COUNT
940 002110 012737 000017 164000 MOV #17, @TRCOM ;DO AN ERASE
941 002116 005037 164000 BR6A: CLR @TRCOM ;FORCE A LOAD PULSE
942 002122 105737 177570 TSTB @SWR ;SEE IF WE WANT TO LOOP
943 002126 100004 BPL BR6B ;ALL DONE
944 002130 105737 164000 TSTB @TRCOM ;SEE IF ERASE IS DONE YET
945 002134 100370 BPL BR6A ;NOT DONE DO ANOTHER LOAD PULSE
946 002138 000756 BR BR6C ;DO IT AGAIN
947 002140 004537 002502 BR6B: JSR R5, @PCL ;DO A CLEAR AND EXIT
948 002144 000137 001510 JMP @#BR1 ;GO BACK TO MAIN
*************** ROUTINE FOR DELAY 10 ***************

BR10: JSR R5, @WETS ; CHECK FOR E.O.T.
       JSR R5, @WFLP ; GET US OFF LOAD POINT
       CLR @TRSTAT ; CLEAR INHIBIT
       MOV @17, @TRCOM ; DO A MAXIMUM ERASE
       MOV #17, #R1 ; DO A MAXIMUM ERASE
       MOV #5000, #R0 ; SET UP COUNTER
       MOV #4, #R1 ; SET TIMING LOOP MULTIPLIER
       DEC R0 ; START COUNTDOWN
       BNE BR10B ; TIMER IS TICKING
       DEC R1 ; ONCE THROUGH TIMING LOOP
       BNE BR10B ; REPEAT LOOP IF MULTIPLIER IS NON ZERO
       CLR @TRWC ; RE ESTABLISH THE ERASE COUNT TO 0
       TSTB @SWR ; SEE IF WE LOOP HERE
       BMI BR10A ; DO IT AGAIN
       JSR R5, @PCL ; DO A POWER CLEAR
       JMP @BR1 ; GET OUT AND GET THE NEXT DELAY DIRECTIVE
** ROUTINE FOR DELAY 11 AND 12 **

BR11: JSR $R5, @REW
      MOV $33, @TRCOM
      WRITE AN I.D.B.
      JSR $R5, @RDY
      CHECK FOR DONE
      JSR $R5, @ERCK
      SEE IF ANY ERRORS UP
      TSTB @SUR
      SEE IF WE WANT TO LOOP
      BMI BR11
      LOOP BACK DO IT AGAIN
      JSR $R5, @PCL
      DO A POWER CLEAR
      JMP @BR1
      GET OUT AND GET NEXT DELAY DIRECTIVE

** ROUTINE FOR DELAY 16 **

BR14: TSTB @TRSTAT
      CHECK FOR EOT UP
      BMI BR14A
      SKIP THE FAST FORWARD
      CLR @TRSTAT
      CLEAR INHIBIT
      MOV @27, @TRCOM
      DO A FAST FORWARD
      JSR $R5, @R DY
      WAIT TILL DONE
      BR14A: CLR @TRSTAT
      CLEAR INHIBIT
      MOV @27, @TRCOM
      TRY A FAST FORWARD, SHOULD PRODUCE ERROR
      JSR $R5, @R DY
      WAIT TILL DONE
      TSTB @SUR
      SEE IF WE LOOP HERE
      BMI BR14A
      YES LOOP HERE
      JSR $R5, @PCL
      DO A POWER CLEAR
      JMP @BR1
      GET OUT DO NEXT DELAY
**ROUTINE FOR DELAY 13**

----------

BR16: JSR R5,REW ; DO A REWIND
               CLR @TRSTAT ; CLEAR INHIBIT
               MOV #17000, @TRWC ; SET UP ERASE COUNT
               JSR R5,TRCOM ; DO AN ERASE
               JSR R5,RDY ; WAIT FOR IT
               JSR R5,REW ; REWIND IT
               JSR R5,ORLP ; GET OFF LOAD POINT LEGALLY
               JSR R5,REW ; REWIND IT
               MOV #3, @TRCOM ; NOW TRY TO WRITE NOTHING, SHOULD ABORT
               MOVC #40, R1 ; SET UP TIME MULTIPLIER
               BR16B: CLR R0 ; CLEAR TIMER
               INC R0 ; TIMES WAISTING
               BNE BR16C ; TIMER RUNNING
               DEC R1 ; -1 FROM MULTIPLIER
               BNE BR16D ; SEE IF DONE YET
               JSR R5,MPCL ; CLEAR ERRORS
               TSTB #SWR ; LOOP ???
               BMI BR16A ; DO IT AGAIN
               JSR R5,MPCL ; POWER CLEAR
               JMP #BR1 ; ALL DONE GET SOME MORE
* SUBROUTINE FOR POWER CLEAR *

**CALL 4000.**

:SET POWER CLEAR

**CALL 4000.**

:WAIT FOR 900 MILI SECONDS

BNE PCL1

:STILL WAITING

RTS R5

:RETURN TO MAIN ROUTINE

* SUBROUTINE FOR READY AND CLEAR INHIBIT *

**CALL 4000.**

:CHECK ON DONE BIT

BPL RDY

:WAIT TILL DONE

RTS R5

:RETURN TO MAIN ROUTINE

* SUBROUTINE TO CHECK FOR ERRORS *

**CALL 4000.**

:SEE IF ERROR IS UP

BPL ERCK2

:NO ERRORS JUMP OUT

MOV #7.,@ITBUF

:GOT AN ERROR RING A BELL

ERCK1:

**CALL 4000.**

:WAIT HERE

BPL ERCK1

:WAIT HERE

JSR R5, @PCL

:CLEAR THE ERROR WITH A POWER CLEAR

ERCK2:

RTS R5

:GO BACK TO MAIN
**SUBROUTINE TO GET OFF LOAD POINT LEGALLY**

```assembly
OFLP: BIT #40,@TRSTAT ; SEE IF LOAD POINT IS UP
BEQ OFLP1 ; NO LP JUMP OUT
CLR @TRSTAT ; CLEAR THE INHIBIT
MOV #55,@TRCOM ; WRITE AN I.D.B.
JSR R5,@RDY ; WAIT FOR READY
JSR R5,ERCK ; CHECK FOR ERRORS
RTS R5 ; GO BACK TO MAIN
```

**SUBROUTINE FOR REWIND AND L.P.**

```assembly
REW: BIT #40,@TRSTAT ; AT LOAD POINT??
BNE REW2 ; YES DON'T NEED REWIND
CLR @TRSTAT ; CLR THE INHIBIT
MOV #21,@TRCOM ; DO A REWIND
JSR R5,@RDY ; WAIT TILL DONE
RTS R5 ; GO BACK
```

**SUBROUTINE FOR E.O.T.**

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

/.=2700
.REPT 1000
.WORD 000777
;ALL 1'S
.\ENDR

/.=3700
.REPT 1000
.WORD 006400
;1'S AND O'S ALTERNATE WORDS
.WORD 000777
;PATTERN = 0 000 000 111 111 111
.\ENDR

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

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

 *************** READ BUFFER AREA ***************

/.=6700
;1000 WORD LOCATIONS RESERVED FOR READ BUFFER

.\END
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALD</td>
<td>000352</td>
<td></td>
</tr>
<tr>
<td>BEGIN</td>
<td>000200</td>
<td></td>
</tr>
<tr>
<td>BEGIN2</td>
<td>000204</td>
<td></td>
</tr>
<tr>
<td>BR1</td>
<td>001510</td>
<td></td>
</tr>
<tr>
<td>BR10</td>
<td>002150</td>
<td></td>
</tr>
<tr>
<td>BR10A</td>
<td>002172</td>
<td></td>
</tr>
<tr>
<td>BR10B</td>
<td>002202</td>
<td></td>
</tr>
<tr>
<td>BR11</td>
<td>002234</td>
<td></td>
</tr>
<tr>
<td>BR14</td>
<td>002274</td>
<td></td>
</tr>
<tr>
<td>BR14A</td>
<td>002320</td>
<td></td>
</tr>
<tr>
<td>BR16</td>
<td>002354</td>
<td></td>
</tr>
<tr>
<td>PR16A</td>
<td>002414</td>
<td></td>
</tr>
<tr>
<td>BR16B</td>
<td>002442</td>
<td></td>
</tr>
<tr>
<td>BR16C</td>
<td>002450</td>
<td></td>
</tr>
<tr>
<td>BR16D</td>
<td>002464</td>
<td></td>
</tr>
<tr>
<td>BR2</td>
<td>001646</td>
<td></td>
</tr>
<tr>
<td>BR2A</td>
<td>001660</td>
<td></td>
</tr>
<tr>
<td>BR3</td>
<td>001702</td>
<td></td>
</tr>
<tr>
<td>BR4</td>
<td>001752</td>
<td></td>
</tr>
<tr>
<td>BR4A</td>
<td>001756</td>
<td></td>
</tr>
<tr>
<td>BR5</td>
<td>002014</td>
<td></td>
</tr>
<tr>
<td>BR5A</td>
<td>002024</td>
<td></td>
</tr>
<tr>
<td>BR6</td>
<td>002070</td>
<td></td>
</tr>
<tr>
<td>BR6A</td>
<td>002116</td>
<td></td>
</tr>
<tr>
<td>BR6B</td>
<td>002140</td>
<td></td>
</tr>
<tr>
<td>BR6C</td>
<td>002074</td>
<td></td>
</tr>
<tr>
<td>CLINH</td>
<td>001150</td>
<td></td>
</tr>
<tr>
<td>COMTEM</td>
<td>000670</td>
<td></td>
</tr>
<tr>
<td>CORBL</td>
<td>000230</td>
<td></td>
</tr>
<tr>
<td>CRES</td>
<td>001106</td>
<td></td>
</tr>
<tr>
<td>DROUTS</td>
<td>001500</td>
<td></td>
</tr>
<tr>
<td>DRIN</td>
<td>001462</td>
<td></td>
</tr>
<tr>
<td>ER</td>
<td>001150</td>
<td></td>
</tr>
<tr>
<td>ERCK</td>
<td>002540</td>
<td></td>
</tr>
<tr>
<td>ERCK1</td>
<td>002554</td>
<td></td>
</tr>
<tr>
<td>ERCK2</td>
<td>002566</td>
<td></td>
</tr>
<tr>
<td>ERDONE</td>
<td>001322</td>
<td></td>
</tr>
<tr>
<td>ERRF</td>
<td>001350</td>
<td></td>
</tr>
<tr>
<td>ERR14</td>
<td>001320</td>
<td></td>
</tr>
<tr>
<td>ERR15</td>
<td>001332</td>
<td></td>
</tr>
<tr>
<td>ERSNCT</td>
<td>000712</td>
<td></td>
</tr>
<tr>
<td>ERSE</td>
<td>001236</td>
<td></td>
</tr>
<tr>
<td>ETS</td>
<td>002654</td>
<td></td>
</tr>
<tr>
<td>ETS1</td>
<td>002666</td>
<td></td>
</tr>
<tr>
<td>EXECUT</td>
<td>001246</td>
<td></td>
</tr>
<tr>
<td>EXTCOR</td>
<td>000700</td>
<td></td>
</tr>
<tr>
<td>FFOR</td>
<td>001154</td>
<td></td>
</tr>
<tr>
<td>FUDONE</td>
<td>001264</td>
<td></td>
</tr>
<tr>
<td>I0B</td>
<td>001156</td>
<td></td>
</tr>
<tr>
<td>IMB</td>
<td>002530</td>
<td></td>
</tr>
<tr>
<td>ILFUT</td>
<td>001344</td>
<td></td>
</tr>
<tr>
<td>ILLCOM</td>
<td>001310</td>
<td></td>
</tr>
<tr>
<td>ILLFUN</td>
<td>001340</td>
<td></td>
</tr>
<tr>
<td>INSMLT</td>
<td>001406</td>
<td></td>
</tr>
<tr>
<td>LEGOPS</td>
<td>0011440</td>
<td></td>
</tr>
<tr>
<td>LOOP</td>
<td>001040</td>
<td></td>
</tr>
</tbody>
</table>

Note: The table represents a cross-reference table with user symbols and their corresponding values. The format suggests it is a part of a larger documentation or software project.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORST</td>
<td>000356</td>
<td></td>
</tr>
<tr>
<td>OFLP</td>
<td>002570</td>
<td></td>
</tr>
<tr>
<td>OFLP1</td>
<td>002622</td>
<td></td>
</tr>
<tr>
<td>OPDEL</td>
<td>001352</td>
<td></td>
</tr>
<tr>
<td>OPDLX</td>
<td>000716</td>
<td></td>
</tr>
<tr>
<td>OPDLY</td>
<td>000714</td>
<td></td>
</tr>
<tr>
<td>OPHUM</td>
<td>000720</td>
<td></td>
</tr>
<tr>
<td>OPTBL</td>
<td>000722</td>
<td></td>
</tr>
<tr>
<td>PASHLI</td>
<td>001432</td>
<td></td>
</tr>
<tr>
<td>PCL</td>
<td>002502</td>
<td></td>
</tr>
<tr>
<td>PCL1</td>
<td>002510</td>
<td></td>
</tr>
<tr>
<td>PSW</td>
<td>001144</td>
<td></td>
</tr>
<tr>
<td>RADD</td>
<td>000706</td>
<td></td>
</tr>
<tr>
<td>RDY</td>
<td>002522</td>
<td></td>
</tr>
<tr>
<td>READ</td>
<td>001220</td>
<td></td>
</tr>
<tr>
<td>REST</td>
<td>001634</td>
<td></td>
</tr>
<tr>
<td>REW</td>
<td>002624</td>
<td></td>
</tr>
<tr>
<td>REWD</td>
<td>001152</td>
<td></td>
</tr>
<tr>
<td>REW2</td>
<td>002634</td>
<td></td>
</tr>
<tr>
<td>SOGB</td>
<td>001162</td>
<td></td>
</tr>
<tr>
<td>SLB</td>
<td>000302</td>
<td></td>
</tr>
<tr>
<td>SLBDOM</td>
<td>000334</td>
<td></td>
</tr>
<tr>
<td>SLB1</td>
<td>000310</td>
<td></td>
</tr>
<tr>
<td>SR</td>
<td>001146</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>001000</td>
<td></td>
</tr>
<tr>
<td>STIM</td>
<td>000672</td>
<td></td>
</tr>
<tr>
<td>SUBSK</td>
<td>000500</td>
<td></td>
</tr>
<tr>
<td>SWR</td>
<td>0007570</td>
<td></td>
</tr>
<tr>
<td>SWRTEM</td>
<td>000664</td>
<td></td>
</tr>
<tr>
<td>TABLE</td>
<td>001346</td>
<td></td>
</tr>
<tr>
<td>TEMP1</td>
<td>000674</td>
<td></td>
</tr>
<tr>
<td>TEMP2</td>
<td>000676</td>
<td></td>
</tr>
<tr>
<td>TIMMUL</td>
<td>000666</td>
<td></td>
</tr>
<tr>
<td>TRBA</td>
<td>164006</td>
<td></td>
</tr>
<tr>
<td>TRCOM</td>
<td>164000</td>
<td></td>
</tr>
<tr>
<td>TRSTAT</td>
<td>164002</td>
<td></td>
</tr>
<tr>
<td>TRWC</td>
<td>164004</td>
<td></td>
</tr>
<tr>
<td>TTBUF</td>
<td>177566</td>
<td></td>
</tr>
<tr>
<td>TTSTAT</td>
<td>177564</td>
<td></td>
</tr>
<tr>
<td>UNIT</td>
<td>000702</td>
<td></td>
</tr>
<tr>
<td>UNNUM</td>
<td>001026</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>001142</td>
<td></td>
</tr>
<tr>
<td>WADDR</td>
<td>000710</td>
<td></td>
</tr>
<tr>
<td>WCNT</td>
<td>000704</td>
<td></td>
</tr>
<tr>
<td>WRITE</td>
<td>001210</td>
<td></td>
</tr>
<tr>
<td>WRIWC</td>
<td>001226</td>
<td></td>
</tr>
<tr>
<td>WTM</td>
<td>001160</td>
<td></td>
</tr>
<tr>
<td>X1</td>
<td>000240</td>
<td></td>
</tr>
<tr>
<td>X2</td>
<td>000244</td>
<td></td>
</tr>
<tr>
<td>X3</td>
<td>000250</td>
<td></td>
</tr>
<tr>
<td>X4</td>
<td>000264</td>
<td></td>
</tr>
<tr>
<td>X5</td>
<td>000267</td>
<td></td>
</tr>
</tbody>
</table>

The cross reference table includes user symbols with corresponding values.
CZTRBD.P11  18-NOV-81 09:16  CROSS REFERENCE TABLE -- USER SYMBOLS

. ABS.  002670    000

ERRORS DETECTED:  0
CZTRBD/I.CZTRBD_SEQ/CRF/SOL/NL:TOC=CZTRBD.P11
RUN-TIME:  12.2 SECONDS
RUN-TIME RATIO:  6/4=1.4
CORE USED:  16K (31 PAGES)